

Wetenschappelijke  
informatie

**JAARGANG : 8**  
**KWARTAAL : 4**

Verschijnt 4 maal per jaar.  
Losse verkoopprijs f 7,-.



"Sinclair Impuls", HET blad voor en door de gebruikers van ALLE Sinclaircomputers - ZX80, ZX81, ZX Spectrum, QL en aanverwanten - wordt uitgegeven door de "HCC Sinclair Gebruikers Groep" (SGG).

IMPULSREDACTIE:

Ed Weijgers  
H Marsmanlaan 29  
2624 TJ Delft

Kees Versluis  
Copernicuslaan 25  
2561 VA Den Haag

Marco Holmer  
J P Coenstraat 61 bis  
3531 EN Utrecht

Jack Raats (eindred)  
Noorddonk 107  
4651 ZD Steenbergen

IMPULSKOPIJ:

Voor OD naar Kees Versluis (OD-hulp: 070-3604185),  
alle andere kopij naar Ed Weijgers of Jack Raats

IMPULSABONNEMENTEN:

Alleen voor HCC-SGG-leden f 25.00 per kalenderjaar  
(voor anderen f 30.00), over te maken op onze SGG-  
postrekening, o.v.v. 'Abonnement Impuls 1991' (al ver-  
schenen nummers ontvangt u bij het volgende nummer).  
Adreswijzigingen: schriftelijk naar het SGG-adres.

SGG-ADRES EN -REKENING:

HCC Sinclair GG  
Postbus 76  
2260 AB Leidschendam

Postrekening 5374525  
tnv HCC Sinclair GG  
te Bunnik

SGG-TELEFOONNUMMER:

Infotelefoon 01670-66845 ma en do, 20-22 uur

Gebruik dit telefoonnummer uitsluitend bij problemen  
met hard- en software of voor de algemene informatie  
over onze SGG, Sinclaircomputers en Impulsartikelen.

SGG-DATABANK "SINCLAIR BOX":

Infotelefoon 01670-66845  
300 1200 2400 1200/75 baud

dagelijks 23-6.00 uur  
za en zo vanaf 20 uur

SGG-ARTIKELN:

HCC (infotel 03403-78788)  
Postbus 149  
3990 DC Houten  
(zie IMPULS 83-35)

DUCBANK (K Versluis)  
via onze SGG-rekening  
o.v.v. gewenste DUCDISKS  
(zie IMPULS 83-33)



Het doet ons genoegen u mee te kunnen delen dat wij hebben besloten om er nog een jaartje mee door te gaan. Daarom ontvangt u bij dit nummer een acceptgirokaart van f 30,- voor uw abonnement in 1991. HCC-SGG-leden mogen weer vijf gulden minder overmaken. Gezien het dalende aantal abonnees is dit een beetje een gok, er zal naar verwachting geld bij moeten. Er zijn echter nog steeds meer SGG-leden dan abonnees en nog veel meer Sinclairgebruikers. Mogen wij u daarom vragen of u wilt proberen daarvoor in aanmerking komende kennissen tot een abonnement en/of een HCC-SGG-lidmaatschap over te halen? U ziet zeker het belang van onze landelijke Sinclairbijeenkomsten en ons blad wel in. Alvast bedankt!

Wij hebben erg ons best gedaan om u weer veel informatie over uw geliefde computer te verschaffen in dit nummer, maar we beseffen dat niet iedereen voldoende aan zijn trekken komt. Deze redactie kan niet overal over schrijven. Er is hulp van anderen nodig, bv van de schrijvers in regionale (of al ter ziele gegane) bladen. Waarom zou iets alleen voor een bepaalde regio interessant zijn?

In dit nummer vindt u artikeltjes over 'echte' RS232-interfaces, dat van de tot voor kort door de SGG verkochte VTX5000-modem en het nu nog beperkt door ons te leveren VTX711-interface (f 25,-, HCC-bestelnummer 5454704; aan de balie als altijd goedkoper.) Software voor beide interfaces wordt en zal worden gepubliceerd. Communicatieprogramma's, oa XCOM 7/8, verschijnen bij IMPULSOFT.

Het eerste programma om 'systeemvreemde schijfjes' te kunnen lezen staat ook in deze IMPULS: BD met OD. Het is de bedoeling om voor alle SP-opslagsystemen onderling zulke programma's te maken die alle soorten diskettes en ook alle soorten files aankunnen. Daarna kunnen we over tekstfiles op QL- en PC-schijfjes denken.

Alle in IMPULS 84 t/m 61 gepubliceerde programma's zijn te downloaden bij "SINCLAIR BOX" (01670-66845) en te koop op DUCDISK-21 en op IMPULSOFT-cassette 15 (f 10,-, HCC-bestelnummer 5454115). Door van 84 naar 61 te gaan was het eenvoudiger om al vervangen versies weg te laten. Een overzicht hiervan komt in IMPULS 91.

Voor de OD-ers is goed nieuws in aantocht. Het enige nadeel tov BD en DD gaat verdwijnen: de relatief geringere snelheid van de diskhandelingen. Victor Vogelpoel is al een heel eind gevorderd met het schrijven van een nieuwe OD-ROM, gebaseerd op en compatibel met versie 2.2. Met alles wat de OD nu al te bieden heeft ontstaat er dan een superieur systeem, waarover u heel eenvoudig zult kunnen beschikken: oude ROM eruit, nieuwe erin. Zodra deze snelle ROM door ons geleverd kan worden leest u dat in dit blad. Houd de volgende IMPULS-nummers dus maar goed in de gaten. In deze IMPULS krijgt u al een andere Opus-Discovery-ontdekking van Victor te zien: hoe u tot vier diskdrives aan kunt sluiten.

Het bestuur van onze SGG en de redactie van IMPULS wensen u een zeer vruchtbaar computerjaar 1991 toe. Tot 19 januari in Houten.

Het bekendste tekenprogramma voor de ZX Spectrum is ongetwijfeld The Art Studio. Van dit programma zijn drie versies uitgebracht:

- The Art Studio - voor SP48 en cassette
- The Extended Art Studio (TEAS) - voor SP48 en Microdrive
- The Advanced Art Studio (TAAS) - voor SP128 en Microdrive

Voor de beide laatste zijn Opus-conversieprogramma's geschreven, door Marcel van Dongen. Deze zijn onlangs weer opgenomen in de DUC-bank. Daarbij behoort ook een LOAD-programma, om steeds opnieuw de 'input device' (cursor, joystick, muis) in te stellen.

Wat zijn de extra's in TAAS ten opzichte van TEAS?

- filemenus bij fill patterns, fonts en brushes,
- scrapbook, om delen van SCREEN\$'s in te bewaren,
- de drie overlays staan in RAM,
- een 42K ramdisk.

Bij Marcells TEAS-conversie worden de overlays altijd van drive 1 geload. Bij de SP128 kan dit van drive 6 gebeuren, de RAMdisk. Om dit te realiseren kunt u het LOAD-programma uitbreiden met de volgende regels en het geheel daarna SAVEN.

HET MERGEPROGRAMMA "Xteasloader"

```

29 IF PEEK CODE "K"=CODE "n" THEN CLEAR VAL "3e3":
   FORMAT PI+PI;"RAMDISK": OPEN #PI;" CAT ";PI+PI:
   LPRINT CHR$ PEEK PI;CHR$ NOT PI;"@";: CLOSE #PI:
   FOR f=SGN PI TO PI: LET a$="overlay"+STR$ f:
   MOVE SGN PI;a$ TO PI+PI;a$: NEXT f: CLEAR VAL "25999"
31 IF PEEK CODE "K"=CODE "n" THEN LET z=PI+PI:
   POKE VAL "37929",z: POKE VAL "38137",z:
   POKE VAL "30684",z: POKE VAL "30688",z:
   POKE VAL "37082",CODE " STEP ":
   POKE VAL "37071",CODE " STEP "

```

In beide regels wordt eerst bepaald of er een SP128 onder handen is. Deze veranderde versie blijft dan ook bruikbaar op een SP48. Regel 29: RAMdisk 6 wordt geFORMAT en 80K groot gemaakt. Daarna worden de drie overlays naar de RAMdisk geMOVED. Er blijft zelfs nog 72K over voor screen\$ en fonts.

Regel 31: zes POKES. De overlays worden nu geload van drive 6 en de 3- en 4-icons in het filemenu verschijnen als 'M' (Memory).

Het gebruik: reset uw SP128, selecteer 128 basic en geef 'usr0'. LOAD dan de aangepaste TEAS-BASIC. Drive 1 is nu niet meer nodig om overlays te LOADen. De RAMdisk kunt u op dezelfde manier gebruiken als drive 1 of 2. Houd er dus rekening mee dat TEAS maar 49 titels van SCREEN\$'s of fonts in zijn catalogus kan bewaren.

Victor Vogelpoel - Hengelsestraat 104-21 - 7514 AK Enschede



Sinds kort biedt onze Gebruikersgroep het VTX711-interface aan. Dit RS232-interface is bedoeld om te werken met twee modems van Engels fabrikaat; de Voyager 7 en de Voyager 11. Aangezien RS232 een universele standaard is kan de VTX711 ook voor andere doeleinden ingezet worden. Bedenk; dit is een KNUTSEL-INTERFACE, dwz dat er wel wat moet worden gesoldeerd om het onderste uit de kan te halen. Wat kunt u verwachten? Een echte (dwz hardwarematige) RS232, snelheden tot 19200 baud en software om ermee te werken.

## HISTORIE

De voorloper van de VTX711 is het Microsource-interface, ontworpen door Henk Scholten. Toen Microsource ophield te bestaan wijzigde hij het wat en verkocht hij het aan de Engelse firma MODEM HOUSE. Het kwam op de markt als VTX711 en had, zoals het Microsource-interface, twee schakelaars voor de baudrate-keuze. Later kwam Henk met nog een update met oa slechts een draaischakelaar, waardoor het aantal te kiezen baudrates enorm toenam. Alleen deze VTX711 MET EEN DRAAISCHAKELAAR gaan wij verder ondersteunen.

## HARDWARE

Het interface is gebouwd met de 6850 ACIA (Asynchronous Communications Interface Adapter) van Motorola. Deze chip is eigenlijk ontworpen voor de 6802-processor, maar kan ook op andere processors worden aangesloten. De 6850 ACIA is zodanig aangesloten dat er vier poorten gebruikt kunnen worden voor de besturing:

- ```
- poort 31 (BIN 00011111) : OUTPUT naar Control Register
- poort 63 (BIN 00111111) : INPUT van Status Register
- poort 95 (BIN 01011111) : OUTPUT naar TX Register
- poort 127 (BIN 01111111) : INPUT van RX Register
```

Omdat de VTX711 niet volledig uitgedecodeerd is zijn de poorten 31 en 63 door elkaar te gebruiken, evenals de poorten 95 en 127. Het gebruik van poort 31 maakt de VTX711 ongeschikt voor samenwerking met de DISCiPLE of een Kempston joystick-interface, maar dat kunnen we eenvoudig veranderen. Ga daartoe aldus te werk:

- Onderbreek het spoor ACIA-pen 9 naar SP-connector A7;
- Verbind ACIA-pen 9 met SP-connector A3.

Na deze modificatie worden de volgende poortadressen gebruikt:

- ```
- poort 183 (BIN 10110111) : OUTPUT naar Control Register
                          : INPUT van Status Register
- poort 247 (BIN 10110111) : OUTPUT naar TX Register
                          : INPUT van RX Register
```

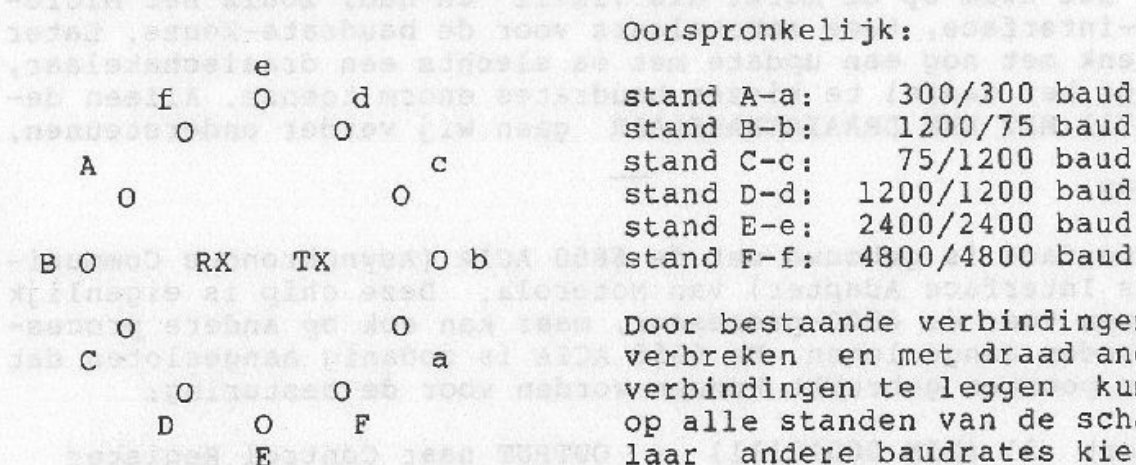
Na deze ingreep moeten in de software de poortadressen aangepast worden. We zullen dat steeds aangeven. Bedenk dat ZX Interface 1 en ons SGG-IF1 eveneens poort 247 gebruiken en zodoende niet gelijktijdig met de gemodificeerde VTX711 aangesloten kunnen zijn.



Nu volgt een interessant punt: de baudrates. De VTX711 heeft een CD4060 als baudrate-generator. De uitgangen van deze chip kunnen worden gebruikt om andere baudrates op te wekken. Kiest u maar:

pen 1 -	75	baud	pen 6 -	2400	baud
pen 2 -	37.5	baud	pen 7 -	19200	baud
pen 3 -	18.75	baud	pen 13 -	600	baud
pen 4 -	4800	baud	pen 14 -	1200	baud
pen 5 -	9600	baud	pen 15 -	300	baud

Hieronder staat een schematisch overzicht van de draaischakelaar vanaf de soldeerzijde. De middelste twee soldeerpunten zijn verbonden met TX en RX van de ACIA. Dit moet zo blijven. De twaalf andere soldeerpunten zijn verbonden met uitgangen van de CD4060.



Houd bij de keuze van de baudrates wel rekening met uw gebruiksdoel. Wanneer u de VTX711 gaat gebruiken met een STACK (-achtig) modem, laat dan de eerste drie oorspronkelijke standen intact.

Nu de connector. Wij raden u aan om de connector gelijk te maken aan die op ZX Interface 1, omdat dit nu eenmaal onze standaard is. Het mooist is natuurlijk de connector te vervangen, maar een verloopkabeltje kan ook. Hieronder staat hoe de oorspronkelijke connector bedraad is, daarnaast die van ZX Interface 1.

VTX711 - 9D-male

2/4	RX	<-
3	TX	->
5	GND	--
7	RTS	->
8	DCD	<-
9	CTS	<-

ZX-IF1 - 9D-female

2	RX	<-
3	TX	->
4	CTS/DSR	<-
5	RTS/DTR	->
7	GND	--
9	9 volt	->

Verder moet u DCD (pen 8) met 5 volt verbinden (anders werkt het niet). Ipv 9 volt op pen 9 voldoet 5 volt ook. Ook kan nog ACIA-pen 7 verbonden worden met SP-connector INT (voor event later te publiceren software). Het solderen en geknutsel is nu ten einde.



## KABELS

Voor het aansluiten van randapparatuur op deze VTX711 kunt u in principe dezelfde kabels gebruiken als voor Interface 1. Echter, voor gebruik met een modem moet CTS (pen 4) altijd hoog gemaakt worden. Dit kan direct met 5 volt of door DCD vanuit het modem.

## SOFTWARE

Geen hardware kan functioneren zonder software. Bij elk RS232-interface behoort (volgens ons) minimaal de volgende software:

- een Viditel-programma (Viditel, Girotel, Comnet, etc),
- een Fido-programma (Fido, Sinclair Box, etc),
- een programma voor file-transfer met XMODEM-protocol
- een 'device driver' voor gebruik vanuit BASIC en MC, voor OUTput en INput (printers, andere computers, etc)

Aan de modem-programma's wordt nog gewerkt. Voor gebruik vanuit BASIC (en MC) volgt nu een 'device driver', die gekoppeld wordt aan het "P"-channel. Eerst voor de niet-gemodificeerde VTX711:

DE MC "C711"><"CODE 65300,188 IN HEXDATAREGELS - KeesV -

1	"2A 4F 5C 11.0F 00 19 11.31 FF 73 23.72 23 11 70"	1019
2	"FF 73 23 72.3E 03 D3 1F.3E 51 D3 1F.C9 CD C5 FF"	3088
3	"20 27 FE A5.38 05 D6 A5.C3 10 0C FD.CB 01 86 FE"	5086
4	"7F 38 02 3E.3F FE 0D 20.07 CD 5D FF.3E 0A 18 09"	6360
5	"FE 20 D8 20.04 FD CB 01.C6 47 CD 54.1F D2 00 0D"	8167
6	"DB 3F E6 0A.FE 02 20 F2.78 D3 5F C9.FD CB 02 9E"	10462
7	"FD CB 37 6E.28 17 ED 7B.3D 5C D1 D1.ED 53 3D 5C"	12550
8	"CD 91 FF 30.FB FE 0D C8.CD 85 0F 18.F3 DB 3F 0F"	14838
9	"38 23 CD 54.1F D2 00 0D.F3 21 80 3E.3E 11 D3 1F"	16259
10	"DB 3F 0F 38.0C 2B 7C B5.20 F6 3E 51.D3 1F AF FB"	18061
11	"C9 3E 51 D3.1F DB 7F CD.C5 FF 20 02.CB BF 37 FB"	20384
12	"C9 47 3A CF.FF F6 20 FE.74 78 C9 62"	22243

Voor de gemodificeerde VTX711 (oa voor Disciple) verkrijgt u de MC "C711mod"><"CODE 65300,188 door de volgende regels te RUNnen:

```
1 FOR f=1 TO 10: READ a,b: POKE a,b: NEXT f: STOP
2 DATA 65323,183,65327,183,65381,183,65426,183,65443,183
3 DATA 65445,183,65457,183,65464,183,65390,247,65466,247
```

Initiëren gaat met RANDOMIZE USR 65300. Voor gebruik als "b"- of "t"-kanaal geeft u POKE 65487, CODE "t" of POKE 65487, CODE "b". Disciple-gebruikers moeten vooraf een "Sysfile" LOADen waarin de printer is uitgeschakeld. Ten slotte nog: wie beschikt over het Viditel-programma (versie voor MS-interface) van Jan Bredenbeek?

Kees Versluis (070-3604185) - Jack Raats (Infotel 01670-66845)



Eerst nog enkele aanvullende opmerkingen bij ons vorige artikel.

Het bleek niet ieder duidelijk te zijn dat "N\$ BESTAAT" en "N\$ ONTSTAAT" koppen boven de twee kolommen rechts zijn. Zonder toevoeging wordt een bestaande file N\$ dus geOPENd voor INput, maar een nieuwe file voor OUTput. Bij IN, EXP (L) en RND (R) moet de file reeds bestaan, anders wordt er gestOPT met een foutmelding.

Bij OUT (L) ontstaat een nieuwe N\$, een evt bestaande verdwijnt. Dit geldt bij RND R,N ook, waarbij bovendien de bytes niet alleen gereserveerd, maar meteen ook definitief toegekend worden. De filelengte kan dus door CLOSE \$S niet meer beperkt worden tot alle tot dan toe naar \$S geschreven bytes. Voeg hier verder aan toe dat N\$, behalve voor OUTput, ook nog voor INput geschikt is.

De opdracht POINT \$S;P kan bij een diskfile pas worden gebruikt, wanneer aan die file reeds definitief een lengte werd toegekend.

In OPUS 1 hebben we het over IN- en OUTput gehad, via naar diskfiles geOPENde streams. Maar dit is niet de enige mogelijkheid. We verzwegen dat het hierbij ging om slechts een der 'channels': "M" (Microdrive, ivm de compatibiliteit met de ZX-IF1-syntaxis). De volledige syntaxis van de gebruikte opdracht luidt namelijk:

OPEN \$S;"M";D;N\$ ...

Hierbij is "M" de kanaalnaam of 'channel identifier'. Deze dient nader gespecificeerd te worden door een 'channel specifier', die bestaat uit een drivenummer (1 t/m 6) en een filenaam (hoogstens 10 tekens) met zonodig een toevoeging. Wij mochten "M" weglaten, aangezien "M" bij de OD nou precies het zg 'default channel' is (waardoor dan ook het volgende scheidingsteken ; kan vervallen).

Andere opdrachten zijn mogelijk door OPEN \$S; te vervangen door: SAVE \*, LOAD \*, MERGE \* en VERIFY \* (onder toevoeging van LINE, CODE, SCREEN\$, DATA () en DATA \$() juist als bij CR-gebruik) en ERASE (zonder "M" is "File not found" mogelijk, met "M" niet).

In een kanaalnaam mag naar believen een hoofdletter of een kleine letter worden gebruikt. Nadere specificatie is niet nodig bij het volgende drietal channels, dat iedere SP altijd reeds kent:

naam	stream	'device' (hard- en software)	tbv
"K" aan #0 en #1		toetsenbord (Keyboard)	IN
		onderscherm	OUT
"S" aan #2		bovenscherm (Screen)	OUT
"P" aan #3		Printer (ZX)	OUT

Deze zijn altijd als aangegeven geOPENd naar hun default-streams #0 t/m #3, tenzij u die streams zelf al naar iets anders OPENde.



## DE ANDERE SPECIALE OD-CHANNELS

naam	spec	device en toelichting	tbv
"D"	D-nr	De gehele disk bij MOVE (zie OPUS 1)	IN/OUT
"B"	geen	OD-paralleelpoort, geen bytesfiltering	IN/OUT
"T"	geen of SRNDb	OD-paralleelpoort, met tekstfiltering IN : hoogste bit wordt 0 OUT: 0 t/m 31 op 13 na worden tegengehouden 128 t/m 164 worden door 63 (?) vervangen 165 t/m 255 worden geëxpandeerd (tokens)	IN/OUT
	s=0:	aan 13 (CR) wordt 10 toegevoegd (default s)	
	s=2:	aan 13 wordt geen 10 (LF) toegevoegd	
	s=1:	AT, en TAB werken als bij de ZX-printer	
	b:	printbreedte, default b=256, toevoeging van CR met evt LF na het b-de teken	
"J"	stand	Joystickpoort, 'aan'/'uit' zetten door FORMAT "J";1 / FORMAT "J";0	IN 31
"#"	S-nr	Stream-channel (doorkoppeling) OPEN #S;#S' ipv OPEN #S;"#";S' mag	IN/OUT
" CAT "	D-nr	CATalogus-channel (zie OPUS 1) niet te verwarren met de CAT-opdracht	IN/OUT
"CODE "	geen	CODE-channel, SP-geheugen: 16K-64K OD-geheugen: 0-16K (zie IMPULS 81-32)	IN/OUT

OPEN #S is mogelijk bij "M", "B", "T", "#", " CAT " en "CODE ".

POINT-opdrachten zijn mogelijk bij een stream S, geopend naar:

channel	pointer na OPEN #S	recordlengte
"M"	record 1	1 (default) of R (dmv RND)
" CAT "	record 2	16 (altijd)
"CODE "	adres 8K (IC0)	1 (altijd, record is adres)

FORMAT-opdrachten zijn mogelijk bij "J" (zie hierboven) en "M":

FORMAT D;N\$ voorziet een disk van tracks met blokken, een ankerblok en een CAT-file, volgens de disksubtabel van D. Deze tabel zit in de OD-ROM of in IC0 mits daartoe een IC6116 geplaatst is. In IC0 wordt deze aangepast aan een disk door deze aan te spreken, bv met CAT D. Daarna wordt een disk in drive D op dezelfde wijze geFORMAT (bv: 2\*40 tracks, 10 sectors, 512 bytes/blok).



In IMPULS 72-09 beschreef Wim Beekman hoe het RS232-interface in het VTX5000-modem ook zelfstandig gebruikt kan worden. Sindsdien gebruik ik het interface met een Hayes-modem. De RS232-signalen, plus 9 volt (via een weerstand van 220 Ohm), heb ik aangesloten op een 9-polige D-connector, net zo bedraad als ZX Interface 1. Bedenk dat weliswaar de pennen overeenkomen met ZX-IF1, maar dat de kwaliteit van de VTX-RS232 vele malen hoger is. Dat komt door de toepassing van een USART voor de parallel-serieel omzetting.

Hieronder geef ik de aansluitingen voor een modemkabel, die ook te gebruiken is op ZX-IF1, SGG-IF1 of gelijkbedraade interfaces.

RS232		MODEM		Merk op dat de interfaceconnector als DTE (Data Terminal Equipment) benoemd is en de 25-polige modemconnector als DCE (Data Communications Equipment). Dit is volgens de regels, maar blijft een lastige zaak, zeker omdat de ZX-IF1-RS232 als DCE beschreven wordt.
RX	2	<----	3 RX	
TX	3	---->	2 TX	
DSR	4	<----	8 DCD	
DTR	5	---->	niet	
GND	7	-----	7 GND	
9v	9	---->	20 DTR	

Vrijwel alle modems hebben op pen 20 een hoog signaal nodig. Wim gaf daarom in zijn artikel een serie POKES om DTR te activeren, maar die zijn bij gebruik van een bovenstaande kabel niet nodig. Als u het VTX-interface gebruikt met een uitgeschakelde VTX-ROM, laat dan de BASIC driemaal OUT 255,32 uitvoeren (IMPULS 73-27).

Alle software voor het VTX5000-modem kan in principe ook werken op een Hayes-modem, maar die software is daar niet altijd op berekend. Hayes-modems moeten immers bediend worden middels Hayes-opdrachten. Hoe kunt u dat doen bij de bekende programma's?

Bij VTXCOM (Beekman) kunt u als volgt de Hayes-opdrachten geven:

- kies "invoeren toegangscodes" en voer de opdrachten in;
- kies "inloggen" en geef EXTEND MODE 6;
- kies "inloggen" en toets DELETE;
- kies "inloggen".

Het is een beetje omslachtig, maar ik ben al blij dat het werkt!

Bij FIDOTERM (Koevoets) kunt u de opdrachten in "terminal mode" geven, steeds afgesloten met ENTER. Het modem echoot alle tekens en geeft na elke opdracht "OK". (Hier blijkt een groot verschil met ZX-IF1. Bij de IF1-versie van FIDOTERM verschijnt de modem-echo niet op het scherm en werken op 2400 baud lukt zelfs niet.)

Ook bij VIDITEL V2.1VTX (Bredenbeek) kunnen de opdrachten ingetoetst worden, maar gebruik niet ENTER ter afsluiting (dat geeft een "#"). De GRAPH-toets werkt hier als CONTROL-toets, dus een CHR\$ 13 verstuurt u met GRAPH M. Alweer een probleem opgelost.

Kees Versluis

-

Copernicuslaan 25

-

2561 VA Den Haag



Marcel van Dongen vermoedde al dat de OD met vier diskdrives zou kunnen werken, omdat het DOS in ROM 2.2(2) daarop voorbereid is. Hij en ik hebben niet geweten dat dit zo gemakkelijk zou zijn! Bedenk dat vier drives een opslagcapaciteit van 4\*720 KB kunnen hebben, dus 2.2 MEGABYTE! (80 tracks DS). Handig bij een BBS bv. En de OD kan zowel drives van 3.5 als van 5.25 inch besturen!

Om vier drives te kunnen gebruiken zijn twee aanpassingen nodig: een van de hard- en een van de software. De eerste, voor de aansluiting van drie of vier drives, wordt hier beschreven. Voor de tweede vindt u in deze IMPULS een programma in een apart artikel "DRIVENUMMERINSTELLING" omdat het ook zonder de aansluiting voor extra drives bruikbaar is. Het werkt echter alleen vanaf OD-ROM-versie 2.2 en past disksubtabellen in het benodigde IC6116 aan.

Op de extra connectors behoeven geen drives aangesloten te zijn, net zoals een standaard-OD slechts een drive behoeft te hebben. Bij drie drives kan drivenummer vier bv als secundaire drive ingesteld worden. Extra drives dienen wel van een eigen voeding te worden voorzien, die van de OD zelf kan er met moeite twee aan! IMPULS 81-19 bevat een schema voor een universele drivevoeding.

#### BENODIGDHEDEN

Een IC van het type van IC15 op de OD-print (zie hierna), doorgaans een 7438 of 74LS38, bij elke electronicazaak verkrijgbaar. Een 'flatcable' met connectors: een voor de shugartconnector op de OD-print en voor elke erop aan te sluiten drive een passende.

#### DE HARDWARE-AANPASSING

Ontkoppel de OD van het net en eventueel aangesloten apparaten. Demonteer de kast zover, dat u de printplaat los daarvan voor u hebt liggen. Het is niet nodig om de transformator- en voedingsdraden los te maken. Ontkoppel maar wel de aanwezige drives, dan hebt u wat meer bewegingsvrijheid (noteer hoe ze aangesloten waren). U moet ook bij de onderkant van de printplaat kunnen komen.

Leg de printplaat zo neer dat de edge- en printerconnectors naar u toe liggen. Lokaliseer IC15: deze vindt u in de buurt van de monitoruitgang. Het type is 7438 of 74LS38, meestal gevolgd door een letter (N). Het nieuwe IC, dat ik vanaf nu IC18 zal noemen, zal bovenop IC15 komen te zitten. Maar eerst een paar controles.

Er zijn meerdere OD-issues. Welke u hebt staat op de printplaat bij de voeding. Meestal staat daar een D of een E. Van issue D verkoopt Kees het schema. Issue E verschilt iets van D, oa qua voeding. Ook heeft issue E een andere IC14 (74LS16) en - belangrijk hier - zijn bij IC15 de pennen 1 en 2 andersom aangesloten.

Van IC15 bufferen twee der vier open-collector-NAND's de drive-selectsignalen. De andere twee bufferen de side- en motorselect.



De driveselect mag pas worden doorgelaten als de disk-controller zijn MO-uitgang (Motor On) hoog heeft gemaakt. Deze uitgang is via een AND-poort verbonden met een van de twee ingangen van de NAND-buffers voor motorselect, driveselect A en B. Het is nu dus logisch dat driveselect C en D ook met dat AND-poortje verbonden worden via een NAND-buffer. Dat gaan we met de nieuwe IC18 doen.

Figuur 1 toont een deel van de OD-printplaat bij issue D en laat tevens de aanpassing zien. Bij issue E zijn slechts de aansluitingen op de pennen 1 en 2 van IC15 verwisseld.

Neem nu het nieuwe IC18 en voer de volgende operatie uit:

- Buig pen 1 (of 2 bij issue E), 3, 5, 6, 8, 9, 10, 11, 12 en 13 naar buiten en knip pen 8 en 11 af (geen contact).
- Leg IC18 bovenop IC15 (gelijkgericht, zie ook figuur 2).
- Soldeer elke NIET verbogen pen vast op de overeenkomstige pen van IC15.
- Verbind pen 1 (of 2 bij issue E) en 5 met resp pen 4 en 5 van IC4 (6821 PIA) dmv stukjes draad.
- Verbind pen 9, 10, 12 en 13 met pen 7 (massa).
- Verbind pen 3 en 6 met resp pen 14 en 6 van de shugart-connector (aan de soldeerzijde van de print, zie figuur 1).

Druk de gemaakte shugartkabel vast op de vrije printconnector en bouw de OD weer op. De kabel kan bv door het gat van de printer-connector naar buiten geleid worden. Op de kabel moeten connectors zitten die op de aan te sluiten drives of op hun kabelconnectors passen. U kunt nog een drive in de OD-kast inbouwen, het is echter ook mogelijk om alle drives buiten de kast te houden.

Vergeet niet om de zg 'jumpers' van de te gebruiken drives in te stellen, elk naar een andere driveselectlijn. Om verwarring met de drivenummers te voorkomen zijn deze lijnen hierboven aangegeven met A, B, C en D. Bij de drives kunt u aanduidingen tegenkomen als DS1, DS2, DS3 en DS4, maar ook wel S0, S1, S2 en S3 oid.

Als u nu zou proberen uw installatie met meer dan twee drives te gebruiken, zou u merken dat er meer drives tegelijk kunnen gaan draaien. Eerst moeten de drivenummers 1, 2, 3 en 4 nog ingesteld worden met het genoemde programma. Zodra u dat gedaan hebt, SAVE dan de inhoud van IC0, want iedere keer als u de OD weer aanzet, of IC0 hebt gereset met USR 14070, moet die file herLOAD worden. Het is natuurlijk handiger om de driveconfiguratie in een nieuwe EPROM te bakken. Dan zijn alle drives altijd direct beschikbaar.

Victor Vogelpoel

Hengelosestraat 104-21 7514 AK Enschede - Telefoon 053-339541







De Spectrum is een prima computer om bestanden op bij te houden, bv met Masterfile. Ook in BASIC is heel goed een 'data base' te maken rond een 2-dimensionale stringarray. Een bekend snelheidsprobleem is dan sorteren. Het onderstaande is een bewerking van de sorteerroutine in MATRICKS op Impulsoftcassette SP13. Van het sorteergedeelte ben ik afgebleven. Ik heb het gedeelte herschreven dat de parameters verwerkt. Deze routine sorteert nu elke 2-dimensionale stringarray, met opave van: een selectie, een sorteersleutel en de sorteervolgorde. De routine is relocatable.

DE MC "Cquicksort"CODE 60000,416 - LJM de Wit - KeesV -

```

1  "2A 0B 5C FD.36 D2 00 23.23 23 23 5E.23 56 E5 D5" 1459
2  "2A 5D 5C E3.22 5D 5C CD.B2 28 DA 2E.1C C1 ED 43" 3344
3  "5D 5C 7E E6.E0 FE C0 C2.20 2A 23 23.23 7E FE 02" 5310
4  "20 F5 23 5E.23 56 ED 53.A8 5B 23 5E.23 56 ED 53" 6986
5  "AA 5B ED 53.BA 5B 23 22.AE 5B E1 23.23 23 23 06" 8549
6  "05 23 23 C5.CD B4 33 C1.23 10 F6 CD.94 1E DD 21" 10384
7  "BC 5B DD 77.00 CD 99 1E.ED 43 B6 5B.CD 99 1E 2A" 12398
8  "AA 5B E5 AF.ED 42 DA 20.2A 0B ED 43.B4 5B E1 ED" 14706
9  "5B B6 5B ED.52 38 EF EB.ED 42 22 B8.5B 28 98 38" 16779
10 "E5 CD 99 1E.2A A8 5B AF.ED 42 38 DA.C5 D1 2A AA" 19067
11 "5B CD A9 30.EB 2A AE 5B.E5 19 22 B2.5B CD 99 1E" 21067
12 "79 B0 28 D9.0B C5 D1 2A.AA 5B CD A9.30 EB E1 19" 23248
13 "22 AE 5B EB.2A B2 5B AF.ED 52 22 AC.5B 28 BE 38" 25170
14 "A5 ED 4B AA.5B 11 00 00.13 ED 42 20.FB ED 53 B0" 27026
15 "5B 2A AA 5B.ED 4B AC 5B.AF ED 42 D0.2A B0 5B 44" 29058
16 "4D CB 28 CB.19 AF ED 42.22 B0 5B 2A.AE 5B C5 54" 30973
17 "5D ED 4B AA.5B 09 E5 09.ED 4B B2 5B.AF ED 42 09" 32954
18 "38 02 60 69.E3 EB AF ED.52 19 EB 30.18 C1 C5 AF" 35066
19 "ED 42 09 30.10 D5 E5 ED.4B B4 5B 09.EB 09 EB ED" 37192
20 "4B B8 5B 18.5C E1 C1 0B.78 B1 20 C2.2A AA 5B 29" 38954
21 "22 AA 5B 18.9C DD CB 00.46 28 11 E1.D1 EB ED 4B" 40961
22 "BA 5B 09 EB.18 BF DD CB.00 46 28 EF.ED 4B BA 5B" 43059
23 "F7 E1 E5 ED.B0 E1 D1 CD.DD 19 1B 62.6B C5 ED 4B" 45799
24 "BA 5B EB 09.EB C1 D5 ED.B8 23 EB ED.4B BA 5B 2A" 48283
25 "61 5C ED B0.D5 ED 5B 61.5C CD E5 19.D1 E1 23 18" 50567
26 "84 1A BE 38.B0 20 BF 13.23 0B 78 B1.20 F3 18 A5" 52196

```

In een BASIC-regel moet de volgende functie-definitie voorkomen:

```
DEF FN s(a$,b,e,l,r,v)=USR adres
```

We zullen eerst de aanwezige parameters de revue laten passeren:

```

a$      : de arraynaam
b & e   : begin & einde van een selectie
l & r   : linker- & rechterslicer van de sorteersleutel
v       : de sorteervolgorde (oneven: a-z, even z-a)
adres   : het LOAD- en aanroepadres

```

In de DEF FN mogen deze parameters ook andere letters hebben.



Daarna kunt u de sorteerroutine aanroepen met (bijvoorbeeld):

```
RANDOMIZE FN s("S$",12,25,10,17,1)
```

De routine zoekt nu in het geheugen naar de array S\$. Van deze array wordt alleen het gedeelte S\$(12) t/m S\$(25) gesorteerd. De sorteersleutel is de slicer 10 TO 17. Dat betekent dat van elke S\$(12) t/m S\$(25) alleen het deel 10 TO 17 wordt beoordeeld voor het sorteren. Met de laatste 1 wordt aangegeven dat de sorteervolgorde 'opklimmend' moet zijn. Moeilijk? Nee, toch.

Ter verduidelijking (hoop ik) volgt hier een swop-sort in BASIC die hetzelfde doet. De werking is goed te volgen, probeer maar. En let ook op overeenkomsten met de bovenstaande functieaanroep.

```
10 FOR a=12 TO 25-1
20   FOR b=a+1 TO 25
30     IF S$(b,10 TO 17)<S$(a,10 TO 17) THEN
        LET Z$=S$(a): LET S$(a)=S$(b): LET S$(b)=Z$
40   NEXT b
50 NEXT a
```

In BETA BASIC 3.0 bestaat de SORT-opdracht. Een goede kennis van mij gebruikt alleen deze sorteer-optie en de rest neemt hij voor lief. Hij gebruikt dus voor bovenstaand voorbeeld deze syntaxis:

```
SORT S$(12 TO 25)(10 TO 17)
```

Mijn kennis kan tevreden zijn. "Cquicksort" is ongeveer net zo snel als BETA BASIC, maar het scheelt hem zo'n 18K aan geheugen. Maar genoeg over BASIC voor dit moment. Ik moet nog enkele opmerkingen kwijt over de sorteerroutine van de vorige bladzijde.

In de routine wordt nagegaan of de parameters in de functie-aanroep wel geldig zijn. Is dat niet het geval, dan verschijnt een toepasselijke foutmelding (Variable not found, Subscript wrong).

Gebruikt u bij de functie-definitie NIET de kortere VAL-notatie. Ik heb vele uren gezocht naar fouten die veroorzaakt werden door  
DEF FN s(...)=USR VAL ".." en DEF FN s(...)=VAL "USR ..".  
Vermeld dus eenvoudig het routine-adres, dwz in gewone cijfers.

Ondanks de lengte is de routine relocatable gebleven. Een plaats boven RAMTOP is aan te bevelen. De routine heeft wel een 21-tal vaste geheugenplaatsen nodig voor 'run time' variabelen. Ik heb daarvoor de printerbuffer gekozen, vanaf adres 23464. Dan kan de routine op alle mogelijke SP-versies werken: 48K, 128K, +2, +3. Als u het niet gelooft, duik dan nog maar eens in de handboeken.

Kees Versluis - Copernicuslaan 25 - 2561 VA Den Haag



Het kopiëren van een aantal opeenvolgende bytes in het geheugen naar een ander adres heet 'BLOKMOVEN'. Bij de OD gaat dit ook in BASIC, door het SAVEN of LOADen van CODE via het "CODE"-kanaal. Dat werkt sneller dan met een FOR-lus en ook in het OD-geheugen.

```
1 INPUT "FROM ";A;" TO ";B;" LEN ";L
2 OPEN #3;"CODE ":POINT #3;B-7: SAVE *#3CODE A,L: CLOSE #3
```

Hierdoor wordt ook een OD-header van 7 bytes voor adres B gezet. Hebt u daar geen plaats voor zo'n header, maar wel voor adres A, dan lukt het nog met een andere regel 2 die LOADt ipv SAVet. Bij de dan benodigde header behoeft slechts de typebyte 3 voor CODE en de in een laag en hoog byte ontbonden lengte L uit SEED ingevuld te worden: LOAD-adres B kan ongesplitst achter CODE komen.

```
2 OPEN #3;"CODE ":POINT #3;A-7: RANDOMIZE L:
  LPRINT CHR$ 3;CHR$ PEEK 23670;CHR$ PEEK 23671:
  POINT #3;A-7: LOAD *#3CODE B: CLOSE #3
```

Deze programma's werken als de MC-instructie LDIR, die A, B en L in de registerparen HL, DE en BC verwacht, maar niet even snel.

#### 'OPLICHTING'

Tijdens het LOADen en SAVEN via het CODE-kanaal wordt er tijdelijk een "#"-kanaal gecreëerd. Hierdoor ontstaat er een 'optil-effect' in oa het BASIC- en VARS-gebied. Voor de LOAD- en SAVE-adressen achter CODE blijkt dit automatisch gecorrigeerd te worden, het POINT-adres daarvoor moet echter met 8 verhoogd worden. We zullen dit toelichten aan de hand van het programmaatje op de bladzijde hiernaast. Wanneer u dat RUNt gebeurt het volgende.

Regel 1 creëert S\$ en zet het adres van S\$(1) in D (uit DEST via SEED, want LET D=FN P(23629) zou D zijn eigen adres toekennen!). Daarna krijgt A het adres van S\$(8), B van S\$(50) en L wordt 25. Regel 2 geeft de eerste 255 elementen van S\$ hun 'plaatsbyte' en regel 9 vertoont de bytewaarden van alle stringelementen.

Geef dan RUN zonder REM achter een van de regelnummers 5 t/m 8.

- regel 5: S\$(8 TO 32) gaat naar S\$(50 TO 74).
- regel 6: hetzelfde, maar met een header vooraan.
- regel 7: S\$(8) komt in elk element van S\$(9 TO 33).
- regel 8: S\$(8 TO 10) gaat zich in S\$(11 TO ) herhalen.

Bij een string kan een header ook zonder LPRINT ingevuld worden. Dat ziet u in regel 8, die 990 (=222+3\*256) als lengte gebruikt. Op deze manier kunt u een string geheel naar believen opvullen.



HET BASICPROGRAMMA "blmovedemo" - EdW -

```

1 DEF FN P(A)=PEEK A+PEEK (A+1)*256:
  CLEAR #: OPEN #3;"CODE ": CLEAR : DIM S$(1E3):
  LET S$(1)="" : RANDOMIZE FN P(23629): LET D=FN P(23670):
  LET A=D+7: LET B=D+49: LET L=25

2 FOR N=1 TO 255: LET S$(N)=CHR$ N: NEXT N

5 REM POINT #3;(B-7)+8: SAVE *#3CODE A,L

6 REM POINT #3;D: RANDOMIZE L:
  LPRINT CHR$ 3;CHR$ PEEK 23670;CHR$ PEEK 23671:
  POINT #3;(A-7)+8: LOAD *#3CODE B

7 REM POINT #3;D: RANDOMIZE L:
  LPRINT CHR$ 3;CHR$ PEEK 23670;CHR$ PEEK 23671:
  POINT #3;D+8: LOAD *#3CODE D+8

8 REM LET S$( TO 3)=CHR$ 3+CHR$ 222+CHR$ 3:
  POINT #3;D+8: LOAD *#3CODE D+10

9 FOR N=1 TO 1E3: PRINT N;TAB 4;CODE S$(N): NEXT N

```

Het opzoeken van een adres in het variabelengeheugen om er bytes vandaan of naartoe te BLOKMOVEn behoeft niet mbv de systeemvariabele DEST te geschieden. Wanneer we de plaats van de variabele kennen, zoals bij S\$ hierboven die direct na CLEAR werd geDIMD, dan kunnen we dat ook mbv VARS (op 23627 en 23628) doen en daarbij de bytes verder tellen die nog voor de elementen staan (zie IMPULS 82-06). Zo zou regel 1 er ook als volgt uit kunnen zien:

```

1 CLEAR #: OPEN #3;"CODE ": CLEAR : DIM S$(1E3):
  LET D=PEEK 23627+PEEK 23628*256+6:
  LET A=D+7: LET B=D+49: LET L=25

```

Op VARS begint nu immers S\$ met 6 bytes voor de elementen: 1 met type en letter (211); 2 met de lengte hierna (1003); 1 met het aantal der dimensies (1); 2 met het aantal der elementen (1E3). Die functie is nu ook overbodig, maar dit vergt eigen rekenwerk.

Uiteraard werd vooraf het CODE-kanaal geOPENd omdat dit ook een optileffect teweeg brengt. Vergeet niet om de 9 bytes daarvan in rekening te brengen wanneer u dat ooit na het adres-PEEKen doet.

PS: Wanneer u de snelheid van deze BLOKMOVE wilt testen voegt u : STOP toe aan regel 2. U kunt dan met CONTINUE doorgaan na uitvoering van de FOR-lus, die heel wat meer tijd blijkt te kosten!

E H F Weijgers

- H Marsmanlaan 29

- 2624 TJ Delft



Dit artikeltje is te beschouwen als een vervolg op dat in IMPULS 62-48 van Jack Raats. Voor ons nieuwste project, het lezen van systeemvreemde schijven, was dat wel nuttig, maar onvoldoende. Omdat het al wat langer geleden is herhalen we het belangrijkste eruit en vullen dat aan met informatie die we ontvingen van Rick Schuitemaker en verkregen uit eigen onderzoek met ... een OD.

## DE TWEE SOORTEN BLOKNUMMERING MET TRACKS EN SECTORS

De BD-directory nummert de tracks van 0 t/m 39 (1\*40 tracks SS), 79 (1\*80 SS of 2\*40 DS) of 159 (2\*80 DS). Bij SS is dit overeenkomstig de fysieke nummering, maar bij DS gaat dit afwisselend:

VOLGENS DE DIRECTORY		DE FYSIEKE NUMMERING
00 02 04 ... 78 ... 158	KANT 1	00 01 02 ... 39 ... 79
01 03 05 ... 79 ... 159	KANT 2	00 01 02 ... 39 ... 79

De BD-directory nummert de blokken van een track van 0 t/m 15, maar fysiek gaat dit van 1 t/m 16 (de zg 'sector offset' is 1).

De zg 'interleave' geeft aan hoeveel blokken verder het volgende blok ligt wanneer beide blokken zich op dezelfde track bevinden. De zg 'skew' geeft aan hoeveel blokken daar nog bijkomen wanneer het volgende blok tot de volgende track behoort. Deze nummervolgorde ivm de verwerkingssnelheid is mij onbekend.

## DE BD-DIRECTORY OP TRACK 0, SECTOR 0 T/M 7: 16 BYTES/FILE

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	
F	I	L	E	N	A	A	M	66	LEN	BLN	N	S	T			BASIC
F	I	L	E	N	A	A	M	67	ADR	LEN	N	S	T			CODE
F	I	L	E	N	A	A	M	68	ADR	LEN	N	S	T			ARRAY
F	I	L	E	N	A	A	M	23	V -	LEN	N	S	T			PRINT

Maximaal 128 files. Ongebruikte filerecords beginnen met een 0, geERASEde met een 1. Files beslaan altijd opeenvolgende blokken.

N: aantal der blokken van 256 bytes       $1 \leq N \leq 255$   
 S: sectornummer van het beginblok       $0 \leq S \leq 15$   
 T: tracknummer van het beginblok       $1 \leq T \leq 39$  of 79 of 159

- CHR\$ 66="B": BASIC      LEN: filelengte      BLN: basiclengte  
 RNR: regelnummer in 2 bytes na 128 170 achter de file zelf  
 (deze 4 bytes worden niet meegeteld in LEN, wel in N)

- CHR\$ 67="C": CODE      ADR: beginadres      LEN: filelengte



- CHR\$ 68="D": ARRAY    ADR: beginadres    LEN: filelengte

Een arrayfile bevat de CODE afkomstig uit het variabelengeheugen (zie IMPULS 82-06), maar zonder de eerste 3 bytes voor L en LEN. De weggelaten LEN is de in de directory vermelde filelengte LEN. De ontbrekende L bevat de naamletter (1 t/m 26 voor a t/m z) en ook het type: 4 (+4\*32 voor numeriek) of 6 (+6\*32 voor string). Met de LEN-berekeningen voor die types uit het IMPULS-artikel is het type terug te vinden. De naamletter van de variabele is niet belangrijk, aangezien die toch vrij te kiezen is bij het LOADen. Arrayfiles duidt de BD (nogal verwarrend) met de D van DATA aan.

- CHR\$ 35="#": PRINT    V    : volgnummer    LEN: filelengte

PRINT-files bestaan uit gelijknamige deelfiles, altijd van 4KB. Tussen deze genummerde deelfiles kunnen andere files voorkomen. Het eerste volgnummer is 0. Op zo'n volgnummer volgt altijd 32. PRINT- of DATA-files bij BD: sequential of random access files.

Er kunnen nog andere gelijknamige files op een BDisk voorkomen, die niet tot dezelfde PRINT-file behoren. Dat zijn dan wel files van verschillend type. De type-aanduidingen in de records van de directory worden voor het onderscheid gebruikt. Bij andere systemen is dat niet mogelijk, maar die werken met namen van 10 ipv 8 tekens. Neem dus het negende teken (B/C/D/#) mee in de namen.

#### DE DISKINFORMATIE OP TRACK 0, SECTOR 8

1 t/m 225 00: ongebruikt

226 & 227 het sector- & tracknummer van het eerste vrije blok  
228 de aantallen der tracks en zijden, aldus gecodeerd:

22: 2\*80 tr (DS 2560 bl)    23: 2\*40 tr (DS 1280 bl)  
24: 1\*80 tr (SS 1280 bl)    25: 1\*40 tr (SS 640 bl)

229 het aantal der files (inclusief evt geERASEde)

230 & 231 het aantal der vrije blokken

232 16: het aantal der blokken per track (sectors)

233 - 256 verschillen per versie

TRACK 0, SECTOR 10 gebruikt de BD voor tijdelijke opslag van de inhoud van de printerbuffer, die zelf dan tot diskbuffer dient.

TRACK 0 blijft verder ongebruikt. De files beginnen op TRACK 1.

Verdere informatie, op- en aanmerkingen zijn als altijd welkom.

E H F Weijgers    -    H Marsmanlaan 29    -    2624 TJ Delft



De Opusgebruikers worden weer verrast met een BASIC-uitbreiding. Met LOAD @ wordt een enkel blok van een disk in het geheugen geload, met SAVE @ gebeurt het omgekeerde. Het zijn heel bruikbare opdrachten, maar er is beslist enige systeemkennis voor vereist. Besef goed dat met name SAVE @ flink wat schade kan aanrichten.

Opdrachten als deze zijn bekend van Betadisk en Disciple. Van de laatste is de syntaxis overgenomen, hoewel het specifieke karakter van de Opus Discovery om een iets andere benadering vraagt. Maar daarover straks wel meer, eerst gaan we de MC installeren. Daar hebt u weer het installatieprogramma "mc>odram" voor nodig, uit IMPULS 74-13. Dat programma moet de volgende regels MERGEN.

HET MERGE-PROGRAMMA "Xl&s:@"

```

60 FOR z=10083 TO c STEP -3:POINT #3;z:
    IF INKEY$#3<>CHR$ 239 THEN NEXT z
61 RANDOMIZE FN g():POINT #3;m+12: LPRINT FN g$():
    POINT #3;z+1: RANDOMIZE m: LPRINT FN g$();
62 FOR z=10083 TO c STEP -3:POINT #3;z:
    IF INKEY$#3<>CHR$ 248 THEN NEXT z
63 RANDOMIZE FN g():POINT #3;m+9: LPRINT FN g$():
    POINT #3;z+1: RANDOMIZE m: LPRINT FN g$();
64 GO TO 70

100 104          : REM tbv MC - zet DATA na elk regelnummer
101 "FE 40 28 0A.FD CB 3A 46.CA 94 02 C3.94 02 D7 79",1985
102 "1C D7 7D 1C.CD 86 04 20.05 D7 82 1C.18 07 DF 28",3428
103 "04 AF D7 28.2D CD 6F 04.D7 94 1E 32.0E 5C D7 99",5144
104 "1E C5 D7 99.1E C5 AF 06.FF F7 0A E5.D7 94 1E F5",7526
105 "06 FF F7 0A.F1 D1 A7 ED.52 20 02 CF.2B F5 FD 35",9815
106 "D4 FD 34 D4.01 00 04 C4.29 0F 3A 74.5C 47 CB 20",11373
107 "0E 00 F1 E1.D1 C3 29 0F" ,12313
108 "",0 : REM tbv opdracht
109 0 : REM tbv plaats
110 21,37,22,5,38,14,39,5,88,229,102,229,0 : REM tbv versie

```

De regels 60-64 koppelen de routine aan het 'operating system'. Regel 64 lijkt misschien overbodig, maar daardoor kan "mc>odram" zonder problemen meerdere MERGE-programma's na elkaar verwerken.

DE SYNTAXIS VAN DEZE EXTRA OPDRACHTEN

```

LOAD @d,b,a      SAVE @d,b,a      d: drivenummer
                                b: bloknummer
                                a: adres
                                i: inquire
                                (ook ,i ipv ;i mag)
                                of
LOAD @d,b,a;i    SAVE @d,b,a;i

```

Geldige drivenummers zijn 1, 2, 3 en 5 voor alle ROM-versies, en ook 4 en 6 (128) vanaf versie 2.2. Ramdisk is dus ook mogelijk.



Zoals gezegd is de syntaxis niet geheel die van de DD. Dat komt omdat de DD met track- en sectornummers en de OD met bloknummers werkt. Ter vermindering van vervelende omrekeningen werd ook hier voor de nummers van de te LOADen of te SAVEN blokken gekozen.

Een standaarddisk (40 tracks met 18 blokken per track) heeft 720 blokken. Het eerste is speciaal: het ankerblok. Hieruit kan het DOS opmaken hoe de disk is geFORMAT: de aantallen van de zijden, de tracks per zijde, de blokken per track en de bytes per blok. Het nummer van dit blok is 65535 ('-1'). Op de blokken 0 t/m 6 staat de 'CATalogue' en de rest, 7 t/m 718, dient voor de files.

Het adres is het begin van de CODE uit of voor een blok. Dit mag in IC0 liggen, omdat de OD-ROM is ingepaged tijdens de executie. Alhoewel er dan ook tijdelijk een "M"-kanaal is geOPENd, hoeft er toch geen rekening gehouden te worden met een 'optileffect', dat wordt bij deze opdrachten geheel automatisch gecompenseerd. De lengte van de CODE is altijd gelijk aan de blok grootte.

Inquire is een vlag. Is deze parameter toegevoegd en niet gelijk aan 0 (defaultwaarde), dan raadpleegt (inquires) het DOS vooraf het ankerblok om de betreffende disksubtabel te 'updaten'. Anders (maar een waarde boven de 255 geeft een foutmelding) gebeurt dit niet. Hierdoor is het mogelijk om met disks van andere systemen te werken, die hebben immers geen OD-ankerblok. U zult de tabel dan wel zelf van de benodigde gegevens moeten voorzien. Sla hiertoe Marcells "disassembly" op, kijk in IMPULS 72-16, of bestudeer de programma's "bdss>od" en "bdss>od" in deze IMPULS.

Een ander probleem is het 'sector starting number': het laagste sectornummer op een track. Voor ODs is dit 0, voor disks van BD, DD of PC is dit 1. Dus als u sector 0 op een track niet kunt vinden ("Disk I/O error"), dan moet dit 'sector starting number' verhoogd worden. Vergeet niet om het na afloop weer te verlagen, want dit gegeven wordt niet 'ge-update' door een diskbenadering.

#### VOORBEELDEN

Eenvoudige blokeditor mbv Tasword II:

Vooraf: LOAD @d,b,32000;1 (dan kun je editen in TW2)  
Daarna: SAVE @d,b,32000;1 (om het blok weer te SAVEN)

Bloktester (voor standaarddiskettes):

FOR f=0 TO 718: LOAD @1,f,50000,NOT f: NEXT f

Als een blok niet te LOADen was verschijnt er "Disk I/O error".

Victor Vogelpoel

Kees Versluis



De meeste lezers van NNN zullen weten wat een BBS is, alleen al omdat ze daar deze NNN vandaan gehaald hebben. Veel mensen zullen ook weten dat er, behalve de normale BBS'en met de scrollende ASCII-teksten, ook nog zoiets bestaat als "videotex", vooral bekend door "Viditel" van ptt-telecom. Je zou videotex-systemen en BBS'en kunnen zien als tegenovergestelden: de meeste gebruikers rekenen zich of tot de videotex-, of tot de BBS-wereld. Discussies daarover kunnen soms hoog oplopen.

Dit artikel gaat over de verschillen tussen de twee systemen, en geeft een beschrijving van videotex aan de hand daarvan. Zowel de verschillen in hardware als de verschillen in software en inhoud komen aan de orde.

Voor de duidelijkheid: in dit artikel bedoel ik met "BBS" een ASCII-terminal-BBS, niet een videotex-systeem. Of dit een juiste keuze was wordt ongetwijfeld de komende maand in NNNLEZER besproken.

#### HARDWARE

Over videotex bestaat een groot misverstand. Je hoort vaak "het zit niet op mijn modem" als excuus om niet met videotex systemen te bellen. Toch is al sinds jaren niet meer noodzakelijk om een V23-modus op je modem te hebben om met videotex-systemen te kunnen werken. Vroeger was V23 (1200/75), zoals Viditel gebruikte, behoorlijk revolutionair: de data verscheen vier keer zo snel op het scherm als bij een acoustisch gekoppeld V21-modem (300/300), dat destijds bij hobbyisten voor BBS'en gebruikelijk was. En dat terwijl de techniek van V23 even eenvoudig en goedkoop was als de techniek van V21. Dat de gebruiker zijn gegevens slechts op typesnelheid kon versturen werd op de koop toe genomen: de meeste informatie gaat immers van BBS naar gebruiker, nietwaar?

Men heeft er wat lang mee gewacht, maar sinds enige tijd is ook bij Viditel, en andere videotex-databanken, voorzichtig de modernisering begonnen. Tegenwoordig zijn alleen nog de kleine prive-videotex-systemen, die vaak draaien op een hobbycomputer, zoals MSX of P2000, alleen bereikbaar met V23. De grotere databanken (die draaien op PC's, of meer in het algemeen: de "meer dan acht"-bitters) zijn met elke modem die ook voor Fido's geschikt is te bereiken. Overigens hebben de >8-bits videotex-systemen vaak meerdere lijnen beschikbaar, meestal via een groepsnummer.

Maar als het "modemverschil" niet meer echt bestaat, wat is dan wel het verschil tussen videotex-systemen en BBS'en? Wat in ieder geval meteen opvalt is de schermopbouw. Het scherm is "bijna precies hetzelfde" als dat van Teletekst: het bestaat uit 24 regels van 40 tekens. Een PC gebruiker zal dat wat weinig vinden, maar omdat alle computers 40 tekens op een regel kunnen zetten ziet een videotex-scherm er bij alle computertypen hetzelfde uit, wanneer de juiste software gebruikt wordt uiteraard.



De videotex-kleurencodes wijken af van de ANSI-kleurencodes die op BBS'en de standaard zijn. Die kleurencodes nemen op het scherm namelijk de positie van een spatie in beslag. Ze hebben effect tot aan het eind van de regel waarop ze staan; aan het begin van elke regel vallen de kleuren automatisch op "wit op zwart" terug. Door een kleurencode te overschrijven, bijvoorbeeld met de code voor een andere kleur, kan een hele regel snel en eenvoudig van kleur veranderd worden.

Verder is het zo dat een BBS meestal met 8 databits zonder pariteit werkt, videotex systemen werken meestal op 7 databits en een even pariteitsbit. Gelukkig stelt de software dat meestal automatisch goed in.

### SYSTEEM

Videotex is gebaseerd op menuschermen, net als veel QuickBBS'en. Maar de videotex-informatie is altijd opgedeeld in pagina's, het scherm scrollt nooit.

Een videotex-systeem is opgehangen aan de menu's: een bestand downloaden of een bericht schrijven is in zekere zin een speciale vorm van een menuscherm. Onderhoud van een videotex-systeem bestaat voor het grootste deel uit het bijwerken van de op de pagina's genoemde informatie, terwijl dat van een QuickBBS-systeem vaak bestaat uit het bijwerken van de file-areas.

Vanuit de hoofdpagina komt u, door steeds een keuze uit een menu te maken, uiteindelijk bij de gewenste informatie terecht. Elke pagina heeft een uniek nummer. Weet u het nummer van een pagina, dan kunt u '\*nummer#' gebruiken om direct naar die pagina te gaan, zonder het hele zoekpad te hoeven doorlopen. De hoofdpagina heeft nummer 0, dus als u onderweg verloren loopt kunt u door '\*0#' in te toetsen altijd terugkomen bij de inlogpagina waarop het hoofdmenu staat.

"Terug naar af" is niet de snelste manier om een pad te vinden, daarvoor zijn handigere commando's. Met keuze '\*#' kan de vorige pagina teruggeroepen worden. Ook is het mogelijk om onderweg een aantal "bladwijzers" te plaatsen: een paginanummer opslaan kan door '\*03x' te kiezen, met voor x een getal tussen 1 en 3. Een opgeslagen paginanummer opvragen kan door '\*04x' in te toetsen. Deze commando's werken op vrijwel alle videotex-systemen.

### DOELGROEPEN

Maar behalve de uiterlijke verschillen is er ook nog een inhoudelijk verschil tussen BBS'en en videotex-systemen. Een BBS verzorgt berichtenuitwisseling (echomail), informatie (Tires), conferencing (chatten), file-transfer (up- en downloaden) en dergelijke. Meestal wordt dit gedaan door computerfreaks.



Vergeleken met BBS'en is het berichtenverkeer tussen gebruikers onderling op een videotex-systeem vrij eenvoudig; berichten moeten worden gesplitst in pagina's van vaak hooguit 20 regels van 40 tekens. Automatische word-wrap is geen gemeengoed in de videotex-wereld. En iets als echomail, berichtenverkeer tussen verschillende videotex-systemen onderling, bestaat nog niet. Voor zijn informatie moet een gebruiker dus meerdere kleinere BBS'en afbellen. Dit verklaart ook de regel "Graag antwoord op dit board" in echomail-berichten; de schrijver ervan is een oud videotex-gebruiker!

De doelgroep van videotex-systemen bestaat uit een breder, minder gespecialiseerd publiek. Men biedt diensten aan als reisinformatie, verstrooiing, telewinkelen en telebankieren. Sinds kort kunnen ook 008-gegevens van de telefoondienst via een videotex-systeem geraadpleegd worden, onder andere voor het vinden van een naam en adres bij een telefoonnummer. Ideaal voor mensen die de kleine advertenties in een plaatselijk krantje lezen. Nadeel is wel dat de interessante videotex-diensten vaak via een duur 06-nummer werken.

De via videotex beschikbare informatie is dus de voornaamste reden waarom u, tevreden BBS-gebruiker, ook eens een voetje op het videotex-ijs zou moeten wagen.

#### TELESOFTWARE

Bij videotex-systemen zijn ook de te downloaden files, daar "telesoftware" genoemd, gebaseerd op pagina's. Het voordeel hiervan is, dat bij elk bestand een reclamepagina gemaakt kan worden die uitnodigt tot downloaden. Op deze pagina staat informatie verborgen die het gebruikersprogramma nodig heeft om de "download" te initiëren.

Tijdens het downloaden zal de videotex-software van de gebruiker automatisch de pagina's opvragen waarop de software in de vorm van tekens is opgeslagen. Omdat in software, en vooral in gecomprimeerde bestanden, ook bytes voorkomen waar geen tekens bij horen (videotex gebruikt maar 7 bits per teken, terwijl een byte 8 bits lang is), worden bepaalde bytes verzonden als twee of drie tekens.

Vanwege deze enorme overhead heeft het downloaden van telesoftware een erg laag rendement: het is ongeveer dubbel zo duur als wat we van Zmodem gewend zijn. Bovendien kan er aan de laatste pagina van een programma een prijs gekoppeld zijn, die automatisch van uw girorekening afgeschreven wordt.

Ook een groot videotex-systeem (ptt-viditel uitgezonderd) heeft vaak niet meer dan 10 of 20 megabytes effectief aan software online, waar vaak per programma apart voor betaald moet worden. Telesoftware is toch interessant om programma's voor een specifieke doelgroep te downloaden: bijvoorbeeld de nieuwste versie



van de mousedriver, of de beschrijving van een speciale hard-disk-controller.

Uploaden naar een videotex-systeem is zeer ongebruikelijk, voornamelijk om een historische reden: V23. Als een videotex-systeem al programma's van gebruikers accepteert, dan gebeurt dat vrijwel altijd via het opsturen van een diskette. Maar de hoofdzaak bij videotex blijft de informatie.

#### GEBRUIKERSSOFTWARE

Zoals al gezegd is voor het raadplegen van een videotex-systeem speciale software nodig, vanwege de afwijkende schermopbouw. Enkele jaren geleden was deze software voor MS-DOS-computers duur, en het assortiment klein. Maar door de initiatieven van enkele actieve gebruikers van IS2000, het videotex-BBS van de Philips Thuiscomputer Club, zijn er enkele programma's geschreven die nu nog steeds gebruikt worden door een groot publiek.

Omdat videotex buiten Nederland niet zoveel toegepast wordt, is er ook niet veel buitenlandse (lees: Amerikaanse) shareware op dit gebied te krijgen. Toch heb ik het een en ander aan goedwerkende videotex-software verzameld en op mijn BBS geplaatst. Veel verschillende programma's hebben sterke punten op hun eigen gebied. Het BBS waar deze software te vinden is, is te bereiken onder telefoonnummer 04990-74979, alleen via V22 en V22bis. De videotex-software staat in de area "DATAcom". Het BBS is aangesloten bij het HCC-net.

Andre van de Wijdeven

BRON : "Nederlands Net Nieuws" - nummer 16 - augustus 1990



Elders in deze IMPULS vindt u een nieuwe versie van "Xpar>ser", voor het RS232-interface op de Opus-paralleelpoort. De ingestelde transmissiesnelheid daarbij is 9600 baud. De baudrate kan weliswaar worden gewijzigd, maar dat is erg omslachtig. Een BASIC-opdracht om de baudrate voor deze routine gemakkelijker te kunnen instellen, is te installeren met onderstaande DATA-regels en het programma "mc>odram" uit IMPULS 74-13. Een prettige bijkomstigheid is dat het dezelfde opdracht als die van ZX Interface 1 is.

HET MERGE-PROGRAMMA "Xformatbt" (DATA achter elk regelnummer):

```

100 164                                     : REM tbv MC
101 "2A 5D 5C E5.D7 FB 24 FD.CB 01 76 28.07 E1 22 5D",1932
102 "5C C3 3D 04.DF 28 29 D7.F1 2B 0B 78.B1 28 02 CF",3644
103 "29 D5 AF 06.06 F7 06 D1.1A F6 20 F5.E5 06 06 F7",5584
104 "06 D1 F1 A7.ED 52 20 D5.FE 74 28 04.FE 62 20 DF",7792
105 "E1 CD 86 04.28 03 D7 8A.1C D7 82 1C.CD 86 04 20",9532
106 "0F D7 82 1C.CD CC EA ED.43 71 27 CD.CF EA 18 07",11696
107 "CD CC EA ED.43 71 27 ED.43 6F 27 C9.CD 6F 04 21",13803
108 "4B 00 D7 7B.00 21 FA EA.FE 6E 28 03.21 FF EA ED",15899
109 "5B 65 5C D7.A9 33 ED B0.ED 53 65 5C.EF 0A 01 27",17833
110 "05 A1 A1 0F.03 38 D7 A2.2D C9 92 05.38 C0 00 92",19402
111 "03 75 D8 9E"                             ,19896
112 " FORMAT ",17                               : REM tbv opdracht
113 85,108,92,111,97,108,118,154,125,159,0 : REM tbv plaats
114 66,37,67,5,77,37,78,5,109,14,110,5,0 : REM tbv versie

```

Voor het instellen van de baudrate kunt u in het vervolg geven:

FORMAT "b";baudrate (B, t of T ipv b maakt niets uit)

Deze opdracht is zelfs nog krachtiger dan die in ZX Interface 1:

- de baudrate-timers komen niet uit een tabel, maar ze worden berekend. Ook minder gangbare baudrates zijn dus toegestaan.
- de SP-versie heeft invloed dmv de formules uit IMPULS 74-37:  
Timer=35E5/26/B-2 (SP48) en Timer=136419/B-2 (SP128).
- "split baudrate" is mogelijk, met deze afwijkende syntaxis:

FORMAT "b";baudrate\_out;baudrate\_in

Dit zal misschien niet vaak gebruikt worden, maar ik wilde u deze optie niet onthouden. Geeft u slechts een baudrate op, dan worden de timers voor OUT en IN aan elkaar gelijk.

Mede door de opkomst van "andere" computers is er een toenemende behoefte aan goede RS232-faciliteiten. Uw Opus is er klaar voor!

Victor Vogelpoel - Hengelosestraat 104-21 - 7514 AK Enschede



Hier volgt het eerste programma dat in het kader van ons project 'het lezen van systeemvreemde schijven' gereedgekomen is. Het is geheel in BASIC geschreven, wat mogelijk was door toepassing van de extra opdracht met LOAD @ van Victor Vogelpoel in het IC6116.

Het kent geen beperkingen. Alle files, van alle vier de BD-types en elke lengte, kunnen op elk soort ODisk worden gezet, met het juiste type: BASIC met eventueel een regelnummer voor autostart; CODE met een beginadres; ARRAYS (A numeriek of A\$); PRINT-files.

Het is bovendien bijzonder simpel en vriendelijk in het gebruik. U krijgt vooraf een genummerde CAT van de BD te zien waaruit u een selectie kunt maken. De verwerking geschiedt daarna volautomatisch, zodat u er dan niet meer bij hoeft te blijven, tenzij u met drive 1 en 3 of 2 en 4 werkt (gebruik dan liever ramdisk).

Op de volgende pagina's vindt u het programma "bdss>od", alsmede regels om daaruit het programma "bdds>od" te krijgen dmv MERGEN. Voor dubbelzijdige BDisks is de bloknummerberekening heel anders ivm zijdwisseling na elke track. Combineren van de grotendeels gelijke programma's lukte niet, maar met MERGEN gaat het prima.

#### - GEBRUIKSAANWIJZING -

Na het LOADen worden de nummers van de drives met de BDisk (kies uit 1 t/m 4) en ODisk (uit 1 t/m 6) gevraagd. Na beantwoording ziet u hoeveel zijden en tracks de BD bevat. Geef dan 0 om door te gaan, 1 indien de BD in een verkeerde drive blijkt te zitten, of STOP als ivm SS/DS het andere conversieprogramma nodig is.

Vervolgens verschijnt de genummerde CATALOGUS van de BD, waarbij van geERASEde files geen namen worden getoond. Noteer de nummers der gewenste files tbv de nu volgende repeterende selectievraag.

Elke vraag toont het aantal der Kbytes van de reeds geselecteerde files, opdat u kunt zien of uw selectie nog op uw ODisk past. Geef na NEXT ? een filenummer en na TO ? een (hoger) nummer als u ook nog daaropvolgende files wilt selecteren, of alleen maar ENTER na TO ? als het bij deze vraag slechts een file moet zijn. Deze selectievraag blijft terugkeren totdat u na NEXT ? alleen maar ENTER geeft, waarop de files in de selectievolgorde naar de OD gekopieerd worden. Bij een vergissing kunt u opnieuw beginnen met de selectie door 0 na NEXT ? te geven. Een selectie mag nummers van geERASEde files bevatten: die worden toch overgeslagen. Gelijksnamige files zijn deelfiles van PRINT-files (random access of sequentieel). Deze dienen opeenvolgend geselecteerd te worden en in dezelfde volgorde als waarin ze bij de CAT getoond werden.

NB: De MC tbv LOAD @ wordt verondersteld in IC0 aanwezig te zijn en tbv de drive met de ODisk erin kan Motor-On-Delay nodig zijn.



HET BASICPROGRAMMA "bdss>od" LINE 555 - EdW - 111190

HET LEZEN EN SCHRIJVEN DER FILEBLOKKEN

```
1 FOR B=B TO B+Z-(2 OR R):          blkn op een na:
    LOAD @D,B,A: LPRINT B$;; NEXT B:      volle blokken
    LOAD @D,B,A:                          laatste blok:
    LPRINT B$( TO R+(256 AND NOT R));: RETURN : meestal deels
    DEF FN I ()=CODE INKEY$#3+CODE INKEY$#3*256
```

HET LEZEN EN TONEN VAN DE BD-CAT

```
3 FOR B=0 TO INT ((N-1)/16):          tot zover nodig
    LOAD @D,B,A+B*256: NEXT B:          naar C$
    PRINT "NOTEER DE NRS DER GEWENSTE FILES""
4 FOR F=1 TO N:                        alle N files
    PRINT F;TAB 4;C$(F, TO 9) AND CODE C$(F)>1, ongeERASED
    CODE C$(F,14): NEXT F:             blokkenaantal
    CLEAR #: LET A=A-273: GO SUB 7      adres van B$(1)
```

HET VERWERKEN DER GESELECTEERDE FILES

```
5 FOR F=1 TO LEN S$:
    LET F$=C$(CODE S$(F)):             bijbeh CAT-rchr
    IF CODE F$<>1 THEN                  mits ongeERASED
        LET B=CODE F$(15)+CODE F$(16)*16: beginblokknummer
        LET L=12-(2 AND F$(9)="B"):      lengtepositie
        LET R=CODE F$(L): LET Z=CODE F$(L+1): rest, hele blkn
        LET L=R+Z*256:                  filelengte
        GO SUB CODE F$(9)*2              type-afh sprong
6    CLOSE #3: NEXT F: GO TO 555        ook als S$=""
```

HET VULLEN VAN DE SELECTIESTRING MET FILENUMMERS

```
7 LET S$="": LET B=0                  blokken in al
8 INPUT "SELECTED",+B/4;" KB",         gekozen files
    " NEXT "; LINE F$;" TO "; LINE T$: nr t/m nr
    IF NOT LEN F$ THEN RETURN          selectie klaar
9 IF NOT LEN T$ THEN LET T$=F$        slechts 1 file
10 LET F=VAL F$: LET T=VAL T$:
    IF F<1 THEN GO TO 7                herbegin
11 FOR F=F TO T AND T<=N:              als F<=T<=N: S$
    LET S$=S$+CHR$ F: LET B=B+CODE C$(F,14): verlengen en/of
    NEXT F: GO TO 8                    anders: terug
    >>
```

Het adres van B\$(1) is 6, dat van C\$(1,1) is 270 hoger dan VARS. Dit wordt berekend bij een geOPENd CAT-kanaal, vanwege eenzelfde optileffect als een geOPENde file later. Het CODE-kanaal tilt 9.



## HET VOORBEREIDEN VAN DE OD-FILES

## - PRINTFILE -

```

70 IF CODE F$(10) THEN                                volgnummer>0:
    OPEN #3;O;F$( TO 9)EXP : GO TO 1                  vervolgfile
71 OPEN #3;O;F$( TO 9): GO TO 1                        nieuwe #-file

```

## - BASICFILE -

```

132 OPEN #3;O;F$( TO 9)RND1,L+7:POINT #3;8: GO SUB 1:  leeg als R=0
    LET T$=B$(R+1 TO ) AND R:                          regelnummer op
    IF R>252 OR NOT R THEN                               volgend blok
        LOAD @D,B+1,A: LET T$=T$+B$                    header
133 POINT #3;1:
    LPRINT CHR$ 0;F$(10 TO 11);T$(3 TO 4);F$(12 TO 13);:
    RETURN

```

## - CODEFILE -

```

134 OPEN #3;O;F$( TO 9):
    LPRINT CHR$ 3;F$(12 TO 13);F$(10 TO 12): GO TO 1    header

```

## - ARRAYFILES &gt; FILETYPE 1 NUMERIEK OF 2 STRING -

```

136 OPEN #3;O;F$( TO 9): LOAD @D,B,A:
    LET N=1+CODE B$*2: LET L=L-N:                      lengte der elem
    LET T$=B$(2 TO N): LET N=1                         T$ met DIM-LENS
137 LET N=N*(CODE T$+CODE T$(2)*256): LET T$=T$(3 TO ):
    IF LEN T$ THEN GO TO 137                            aantal der elem
138 LPRINT CHR$ (2 OR L>N);F$(12 TO 14);               header met var-
    CHR$ (128+(65 OR L>N)),,: GO TO 1                   type 4 of 6

```

## HET INSTELLEN VAN DE OD-DISKABEL MBV BD-INFOBLOK 8

```

555>CLEAR #: CLEAR : DIM B$(256): DIM C$(128,16):  tbv BD-blok
    INPUT "BD IN ";D,"OD IN ";O: OPEN #4;" CAT ";O:   en -CAT
    OPEN #3;"CODE ":POINT #3;FN I()+10:POINT #3;FN I()
556 LET S=CODE INKEY$#3: LET A=FN I():               zoek adres van
    IF S<>D THEN GO TO 556                           disksubtabel
557 POINT #3;A+1: LPRINT CHR$ 16,:                   16 blkn / track
    LET W=CODE INKEY$#3: LET Z=64+W-INT (W/16)*16:    blgr 256
    POINT #3;A+2: LPRINT CHR$ Z;CHR$ 1;CHR$ 0;CHR$ 0;  0 anker
558 POINT #3;A: LET A=PEEK 23627+PEEK 23628*256+270:  adres C$
    LOAD @D,8,A-264: LET T=CODE B$(228):              T uit 22 t/m 25
    LET S=T<24: LET T=40*(2 OR T=23 OR T=25):         S+1 kanten
    LPRINT CHR$ T;CHR$ 16;CHR$ (Z+S*16);:             T tracks / kant
    LET N=CODE B$(229): LET Z=T*16-8                  aantal vd files
559 INPUT ",," FORMAT BD: ";1+S;"*";+T              en Z tbv DS
    "'0 CONTINUE / 1 RETURN ";L:                      evt STOP & LOAD
    GO TO 3+(552 AND L)                                ander programma

```



MERGEREGELS VOOR HET BASICPROGRAMMA "bdds>od" LINE 555 - EdW -

```

1 FOR B=1 TO B-NOT R:
  LOAD @D,T+W+S,A: LPRINT B$;: LET S=S+1:
  IF S=16 THEN                                kantwisseling
    LET T=T+8: LET W=Z AND NOT W: LET S=0
2 NEXT B:                                       laatste blok
LOAD @D,T+W+S,A: LPRINT B$( TO R+(256 AND NOT R));:
RETURN : DEF FN I()=CODE INKEY$#3+CODE INKEY$#3*256

5 FOR F=1 TO LEN S$:
  LET F$=C$(CODE S$(F)):
  IF CODE F$<>1 THEN
    LET S=CODE F$(15): LET T=CODE F$(16)*8: van beginblok
    LET W=Z AND INT (T/16)*16<>T:          tr-wisselterm
    LET L=12-(2 AND F$(9))="B":           lengtepositie
    LET R=CODE F$(L): LET B=CODE F$(L+1): rest, blkkn
    LET L=R+B*256: GO SUB CODE F$(9)*2     filelengte

132 OPEN #3;O:F$( TO 9)RND1,L+7:POINT #3;8: GO SUB 1:
  LET T$=B$(R+1 TO ) AND R:
  IF R>252 OR NOT R THEN
    LOAD @D,T+W+S+1+((Z AND NOT W)-W-8 AND S=15),A:
    LET T$=T$+B$

136 OPEN #3;O:F$( TO 9): LOAD @D,T+W+S,A:
  LET N=1+CODE B$*2: LET L=L-N: LET T$=B$(2 TO N): LET N=1

```

GEZIEN ALS DS-OPUSDISK

GEZIEN ALS DS-BETADISK

t (tracknr als  $B=t*16+S$ ):

T (volgens directory):

0	1	2	...	39	...	79
40	41	42	...	79	of	
80	81	82	..... 159			

0	2	4	....	78	....	158
1	3	5	....	79	....	159
(NB op kant 1 is T even)						

B (bloknummer op tr bij  $S=0$ ):

T met 8 vermenigvuldigd:

0	16	32	....	624	..	1264
640	656	672	...	1264	of	
1280	1296	1312	..... 2544			

0	16	32	...	624	...	1264
8	24	40	...	632	...	1272
						40 of 80 tr

Dus  $B=T*8+W+S$  met  $W=$  0 (kant 1) en  $Z=$  632=40\*16-8  
 Z (kant 2) 1272=80\*16-8

E H F Weijgers - H Marsmanlaan 29 - TJ 2624 Delft



In IMPULS 63-22 schreef Rudie Aalders een artikel over rekenroutines voor vermenigvuldigen en delen. Tijdens een gevolgde studie ontdekte ik nog een ander algoritme (oplossingsmethode) voor het vermenigvuldigen van getallen. Hierbij is de uitvoeringstijd bijna onafhankelijk van het getal waarmee wordt vermenigvuldigd (de vermenigvuldiger). Deze methode kan zowel bij unsigned (tekenloze) als bij signed (2-complement) getallen worden toegepast en heet: "vermenigvuldigen door schuiven en optellen" (VSOP).

```
Voorbeeld met 4-bits          1011 vermenigvuldigtal
'unsigned' getallen:          1101 vermenigvuldiger
                               - - *
                               1011
                               0000.
                               1011..
                               1011...
                               - - +
                               10001111 resultaat
```

Belangrijke punten:

- Het resultaat staat in 8 bits, dat is dus tweemaal zo lang.
- Het vermenigvuldigtal wordt telkens naar links geschoven en opgeteld als het betreffende bit in de vermenigvuldiger 1 is.

Een algoritme voor Unsigned Numbers:

UNVSOP

DE <- GETAL1 (vermenigvuldigtal, 8 bits)	
A <- GETAL2 (vermenigvuldiger)	
HL <- 0 (somregister)	
B <- 8 (bit-teller)	
	schuif bit van vermenigvuldiger in de carry flag
	carry flag = 1 ?
	JA NEE
	tel vermenigvuldigtal op bij somregister
	schuif vermenigvuldigtal 1 plaats naar links
	trek 1 af van de bit-teller
TOT	bit-teller = 0 (8 maal gedaan)
RETURN	



We zullen nu het voorgaande programmastructuurdiagram (PSD) gaan omzetten in assembleertaal. Het executieadres daarbij is 40002.

```

GETAL1 EQU 40000 ;8-bits vermenigvuldigtal
GETAL2 EQU 40001 ;8-bits vermenigvuldiger

ORG 40002

UNVSOP LD HL,GETAL1 ;Init datapointer
LD E,(HL) ;DE <- vermenigvuldigtal
LD D,0 ;
INC HL ;
LD A,(HL) ;A <- vermenigvuldiger
LD HL,0 ;Clear somregister
LD B,8 ;Init bit-teller

LOOP SRL A ;b0 vermenigvuldiger naar CY-flag
JR NC,SCHUIF ;CY-FLAG = 0 alleen schuiven
ADD HL,DE ;CY-FLAG = 1 ook optellen

SCHUIF EX DE,HL ;Schuif vermenigvuldigtal 1 plaatst
ADD HL,HL ;naar links (vermenigvuldig met 2)
EX DE,HL ;

DJNZ LOOP ;Herhaal 8* (B <- B - 1 tot B = 0)

RET ;HL = getal1 * getal 2

```

De maximale executietijd van deze routine (als de vermenigvuldiger = #FF) is 513 T-states. Het minimale aantal T-states (als de vermenigvuldiger = #00) is 465. Met andere woorden: de snelheid is bijna niet afhankelijk van de vermenigvuldiger. Als we deze routine vergelijken met routine 3 van Rudie, dan zien we dat de hier besproken routine sneller is als de vermenigvuldiger groter is dan 129. Toch past Rudie bijna dezelfde algoritme toe.

We kunnen de tijd bekorten door in de herhalingslus te testen of de inhoud van A nul geworden is. Als nu de vermenigvuldiger bv. 0 of 1 is, dan wordt de lus maar eenmaal doorlopen, wat resulteert in 109 of 115 T-states. Dit betekent evenwel dat de waarde van de vermenigvuldiger meer invloed krijgt op de uitvoeringstijd van de routine. Dat is dus bij minder dan 8 maal schuiven.

We krijgen dan:

```

.....
AND A ;Vermenigvuldiger 0 ?
JR Z,EINDE ;Zo ja, verlaat routine.

DJNZ LOOP ;Herhaal 8 maal

EINDE RET ;HL = getal1 * getal 2

```



De MC is nog wat meer te versnellen door absolute spronginstructies (JP Z,...) te gebruiken in plaats van relatieve (JR Z,...).

We zullen nu een voorbeeld bespreken met 8-bits signed numbers. Als we met signed numbers werken moeten we, om problemen bij het resultaat te voorkomen, ervoor zorgen dat de woordlengte van de twee getallen minimaal gelijk is aan de woordlengte van het resultaat. Dit doen we door sign-extension toe te passen van 8-bits naar 16-bits. Sign-extension betekent: het tekenbit (b7) herhaald kopiëren totdat er een 16-bits resultaat ontstaat.

Bijvoorbeeld:

+3 in 2-compl: 00000011 na sign-extension: 0000000000000011  
-3 in 2-compl: 11111101 na sign-extension: 1111111111111101

We laten de routine zelf deze sign-extension uitvoeren, zodat we nogmaals de geheugenplaatsen GETAL1 en GETAL2 kunnen gebruiken.

Een algoritme voor Signed Numbers (2-complement):

SNVSOP

DE <- GETAL1 (vermenigvuldigtal, 8 bits)	
sign-extension tot 16 bits	
BC <- GETAL2 (vermenigvuldiger, 8 bits)	
sign-extension tot 16 bits	
HL <- 0 (somregister)	
A <- 16 (bit-teller)	
schuif bit van vermenigvuldiger in de carry flag	
carry flag = 1 ?	
JA	NEE
tel vermenigvuldigtal op bij somregister	
schuif vermenigvuldigtal 1 plaats naar links	
trek 1 af van de bit-teller	
TOT bit-teller = 0 (16 maal gedaan)	
RETURN	



We gaan nu het vorenstaande PSD weer omzetten in assembleertaal.

```

GETAL1 EQU 40000 ;8-bits vermenigvuldigtal
GETAL2 EQU 40001 ;8-bits vermenigvuldiger

ORG 40002

SNVSOP LD HL,GETAL1 ;Init datapointer
LD E,(HL) ;DE <- vermenigvuldigtal
BIT 7,E ;Test sign-bit
LD D,#FF ;b7 = 1 sign-extension met 1
JR NZ,NEXT ;
INC D ;b7 = 0 sign-extension met 0

NEXT INC HL ;
LD C,(HL) ;BC <- vermenigvuldiger
BIT 7,C ;Test sign-bit
LD B,#FF ;b7 = 1 sign-extension met 1
JR NZ,TELOP ;
INC B ;b7 = 0 sign-extension met 0

TELOP LD HL,0 ;Clear somregister
LD A,16 ;Init bit-teller

LOOP SRL B ;b0 vermenigvuldiger naar CY-flag
RR C ;b7 van sign-ext = b0 vermenigvuld
JR NC,SCHUIF ;CY-FLAG = 0 alleen schuiven
ADD HL,DE ;CY-FLAG = 1 ook optellen

SCHUIF EX DE,HL ;Schuif vermenigvuldigtal 1 plaats
ADD HL,HL ;naar links (vermenigvuldig met 2)
EX DE,HL ;
DEC A ;Verminder de teller met 1
JR NZ,LOOP ;Herhaal 16 maal (tot teller = 0)

RET ;HL = getal1 * getal 2

```

De uitvoeringstijd varieert van 1102 (vermenigvuldiger en vermenigvuldigtal zijn beide #00) tot 1203 T-states (de beide waarden zijn #FF). Wederom zien we tijden die elkaar niet veel ontlopen. Deze routines zijn dus goed te gebruiken in tijdgevoelige situaties. Bij de berekeningen is het aantal T-states van een eventuele CALL- en de RET-instructie weggelaten.

Natuurlijk kunnen we ook in deze routine de tijd verkorten door een test op te nemen die bepaalt of de vermenigvuldiger al 0 is voordat 16-maal de schuifinstructies zijn uitgevoerd. In dit geval moet dus bekeken worden of het BC-register al 0 geworden is.

John Leyenaar - J de Bosch-Kemperstraat 49 - 3207 EA Spijkenisse



De volgende programma's zijn niet alleen te gebruiken om daarmee teksten te lezen, maar ook om deze zelf te lezen als voorbeelden bij hetgeen in de artikeltjes van de OPUS-serie wordt behandeld.

Ze komen het best tot hun recht met de MC voor 64 tekens per regel van DUCDISK 20, maar dat is geen noodzaak. Geef bij deze MC wel PRINT CHR\$ 24;CHR\$ 22;CHR\$ 5 voor 22 regels met wel Scroll?.

Kiest u een file met een ongeschikte inhoud, MC bv, dan krijgt u onzin op uw scherm en mogelijk een foutmelding. Dat is niet erg, herstart dan met RETURN (RUN bij "lees:su"). Dat kunt u ook doen nadat u zelf BREAK gaf omdat u al genoeg van de file had gezien.

Weet u niet (meer) hoe het zit met fileheaders en al die soorten tekstfiles? Dat vindt u nog eens aan het einde van dit artikel.

HET BASICPROGRAMMA "lees:su" LINE 1 - EdW -

```
1>CLOSE #3: INPUT "NAAM";N$;"DRIVE",D:
  OPEN #3;D;N$:POINT #3;8 OR INKEY$#3<>CHR$ 3:
  CLEAR : MOVE #3 TO #2: GO TO 1
```

Nadat stream 3 werd vrijgemaakt wordt er om een filenaam en een drivenummer gevraagd. Dan wordt de bestaande tekstfile geOPENd, dus alleen voor lezen. Is het een CODE-file, dan moet de header overgeslagen worden om bij het tekstbegin te komen. Je zou daarvoor bv 7 keer LET I\$=INKEY\$#3 kunnen uitvoeren, desnoods met een FOR-lus. POINT #3;8 zet de pointer echter meteen goed (de recordlengte is 1). Zit die s- of u-tekst in een DATA-file, dan is er geen header. Daartoe staat er 8 OR INKEY\$#3<>CHR\$ 3 : geen CHR\$ 3 voorop (record 1 na OPENen) maakt deze expressie eerst 1, waarna POINT de pointer weer terug zet (was tgv INKEY\$#3 al 2). CLEAR wist het scherm en de variabelen. MOVE gebruikt nl altijd zoveel ruimte tussen BASIC en RAMTOP als mogelijk (en nodig) is. Toepassing van MOVE maakt dit programma sneller dan de volgende. Via #3 wordt de tekst nu naar #2 (aan "S": Screen) gEMOVEd. Tot slot wordt #3 geCLOSEd en kunt u doorgaan of EDIT STOP geven.

HET BASICPROGRAMMA "lees:vd" LINE 2 - EdW -

```
1 LPRINT INKEY$#3 AND 0;: IF USR 432 THEN
  INPUT #3; LINE I$: PRINT I$: GO TO 1

2>CLOSE #3: INPUT "NAAM",N$;"DRIVE",D:
  OPEN #3;D;N$:POINT #3;8 OR INKEY$#3<>CHR$ 3: CLEAR :
  INPUT #3; LINE I$: PRINT I$: GO SUB 1: GO TO 2
```

INPUT leest N\$ - en vult I\$ - tot de volgende CR (niet naar I\$). LINE is nodig om te voorkomen dat dit lezen voortijdig beëindigd



wordt door een stringafsluiter " (di nutteloos, mi een SP-fout). Hiermee zou elke string met LF beginnen, op de eerste na. Daarom leest en PRINT regel 2 de eerste tekstregel gewoon, maar regel 1 leest die LF van de volgende tekstregels eerst 'weg' met INKEY\$. Om niet na de laatste LF de foutmelding "End of file" te krijgen wordt USR 432 gebruikt. Die wordt nl 0, zodra de pointer voorbij de file komt. Dit werkt echter slechts na een PRINT #3-opdracht. Maar die mag niets schrijven. Dat kan ook niet, dat zou ook weer een foutmelding geven omdat #3 slechts voor INPUT werd geopend. Daartoe wordt het PRINT-item mbv AND 0 leeg gemaakt. Dit had ook in twee keer gekund: LET I\$=INKEY\$#3: LPRINT ; (langer, trager). Files per regel lezen met INPUT gaat trager dan ze in hun geheel MOVEN. Voor meer snelheid staat de vaakst doorlopen lus vooraan en worden er zo weinig mogelijk variabelen gebruikt (na CLEAR).

HET BASICPROGRAMMA "lees:stuvw" LINE 6 - EdW -

```

1 LET I$=INKEY$#3: IF I$>=" " THEN PRINT I$;; GO TO 1
2 IF I$=CHR$ 13 OR I$=CHR$ 0 THEN LET V=1: PRINT : GO TO 1
3 IF I$="" THEN RETURN
4 IF V THEN LET V=0: GO TO 1
5 PRINT : GO TO 1

6>CLOSE #3: INPUT "NAAM",N$;"DRIVE",D: OPEN #3;D;N$: CLEAR :
POINT #3;8 OR INKEY$ #3<>CHR$ 3: LET V=0: GO SUB 1: GO TO 6

```

Hiermee kunt u tekstfiles lezen van alle typen (tuwvs), zowel in CODE- als in DATA-vorm. We kunnen nu geen INPUT # gebruiken daar niet alle typen CR bevatten. Bovendien moeten die verschillende regelscheidingen herkend en verschillend verwerkt worden. Daarom moeten alle tekens nu apart gelezen worden met INKEY\$#. Dit gaat uiteraard langzamer dan hele regels tegelijk lezen met INPUT #.

Regel 1 leest elke teken en zet het in I\$. Omdat de eerste regel het snelst toegankelijk is bevat deze de vaakst doorlopen lus, waarmee de 'gewone' tekens gePRINT worden (alle bij een s-file).

Regel 2 begint een nieuwe regel na CHR\$ 13 (uud) of CHR\$ 0 (t), en zet daarbij bovendien een vlag op 'aan' (V=1) tbv een v-file. Aan het begin werd deze vlag op 'uit' (V=0) gezet door regel 6.

Regel 3 geeft een RETURN naar beginregel 6 zodra I\$ leeg blijkt te zijn tgv EOF. Bij INKEY\$# is er geen test mbv USR 432 nodig.

Regel 4 wordt alleen bereikt als I\$=CHR\$ 10. Bij een v-file ging daaraan een CHR\$ 13 vooraf die de vlag op 'aan' zette, maar een w-file bevat geen CHR\$ 13 of CHR\$ 0: de vlag staat nooit 'aan'. Hier wordt alleen een vlag die 'aan' staat weer op 'uit' gezet.

Regel 5 begint aan een nieuwe regel, alleen bij een w-file dus.



Om toch gebruik te kunnen maken van MOVE, de snelste opdracht, is MC nodig die de tekens filtert voor ze naar het scherm gaan. Nadat een eenvoudigere methode om scheidingstekens te verwerken was bedacht die zonder vlag werkt heeft Kees Versluis die MC geschreven. Hier volgen de beide delen op de gebruikelijke wijze.

HET BASICPROGRAMMA "read:txt" LINE 3 - EdW -

```
1 CLOSE #3: INPUT "NAAM",N$:"DRIVE",D: OPEN #3;D;N$:
  LPRINT ;: LET L=USR 432: LET T=8 OR INKEY$#3<>CHR$ 3:
  POINT #3;T: LET V=CODE INKEY$#3:
  IF V THEN
    POINT #3;L: LET V=CODE INKEY$#3
2 POKE 65403,V:POINT #3;T: CLEAR :
  RANDOMIZE USR USR "A": MOVE #3 TO #2: GO TO 1

3>LOAD *;"Cread:txt"CODE : RUN
```

DE MC "Cread:txt"CODE 65368,42 IN HEXDATAREGELS - KV -

1	"3E 02 CD 01.16 2A 51 5C.5E 23 56 EB.01 76 FF A7"	1498
2	"ED 42 C8 EB.56 70 2B 5E.71 ED 53 80.FF C9 FE 20"	3874
3	"30 05 FE 0A.CO 3E 0D C3.00 01"	4654

Eerst bepalen we welke byte (V) we gaan gebruiken om telkens een nieuwe regel te beginnen. Bij type t is dat 0, de eerste tekst-byte. Anders moeten we naar de laatste byte van de file kijken: 13 bij type u, 10 bij type w of type v en 32 of meer bij type s.

Regel 1 slaat na het OPENen de filelengte uit USR 432 op in L, tevens de positie van de laatste tekstbyte, en de positie van de eerste tekstbyte in T. Mbv T wordt de CODE van die eerste tekstbyte in V gezet. Wanneer het geen t-file blijkt te zijn (V<>0), dan wordt mbv L de CODE van de laatste tekstbyte in V gezet.

Regel 2 POKet V in de MC en zet de pointer op het tekstbegin T. Na een CLEAR wordt de MC aangeroepen en vindt de MOVE plaats.

Door het aanroepen van deze MC wordt stream 2 omgeleid via een filter, dat alle bytes boven de 31 doorlaat en alle andere niet. Elke tegengehouden byte wordt vergeleken met de gePOKete byte V, en alleen als deze overeenstemmen wordt er een 13 doorgestuurd. (Bij een s-file gaat alles door, dat V>=32 is dus onbelangrijk.)

Indien u de MC voor 64 tekens per regel wilt gebruiken, dan moet die geLOAD en aangeroepen worden voordat u dit programma LOADt.

Wanneer het lezen afgebroken is kunt u altijd herstarten met RUN (dus niet met RETURN). Na afloop behoeft de #2-omleiding alleen ongedaan gemaakt te worden wanneer er gebruik gemaakt werd van



de MC voor 64 tekens per regel. In dat geval kunt u die MC nogmaals aanroepen, of beeindigen met CLEAR #(2), CLOSE #2 of NEW. Een druk op de resetknop ligt hier echter wat meer voor de hand.

#### DE OPBOUW VAN DE 7-BYTES FILEHEADERS BIJ N-FILES OP DISK

0	FLN	RNR	BLN	.....	BASIC ... (VARS) ..	RNR: regelnummer
1	FLN	-	N	- -	NUMERIEK ARRAY ....	BLN: basiclengte
2	FLN	-	S	- -	STRINGARRAY .....	ADR: beginadres
3	FLN	ADR	- -	.....	CODE .....	FLN: filelengte

achter de header

N: 32+CODE (kl letter)    S: 96+CODE (kl letter)    -: ongebruikt

#### DE OPBOUW BIJ DE TEKSTTYPEN (CODE MET EN DATA ZONDER HEADER)

TYPE	IEDERE REGEL	GEBRUIK
s	.....	SW, TW2 (64 tekens per regel)
t	0 .....	tape-TW3
u	..... 13	upload-messages, bestanden
v d	..... 13 10	TW3 (d is de DATA-vorm van v-CODE)
w	..... 10	PC

Op de puntjes bytes voor 'gewone' tekens: 32 (spatie) en hoger. CHR\$ 13 heet 'Carriage Return' (CR) en CHR\$ 10 'Linefeed' (LF).

In deel 2 krijgt u programma's waarmee u de LISTing kunt bekijken van iedere BASIC-file zonder deze te hoeven LOADen, alsmede een combinatie om zowel tekst- als BASIC-files te kunnen lezen. Wilt u dit alvast zelf proberen? Hier volgt nog wat informatie.

#### DE OPBOUW VAN DE REGELS VAN EEN BASICPROGRAMMA

RNR	LEN	...	opdracht(en)	...	13	RNR: regelnummer
h	l	l	h	<	LEN	>
						LEN: regellengte - 4

Achter ieder getal in cijfers staat: 14 en de FPR/INT (5 bytes). Bytes die daarbuiten niet voorkomen: 8 t/m 12, 15 en 21 t/m 31. Achter 16 (INK) en 17 (PAPER) staat een der kleurbytes 0 t/m 7. Achter 18 (FLASH), 19 (BRIGHT) en 20 (INVERSE) staat een 0 (uit) of een 1 (aan). Achter 16 t/m 20 geven zulke bijbehorende kleur- of toestandbytes geen vraagtekens op het scherm bij het PRINTen. De bytes 21 (OVER), 22 (AT) en 23 (TAB) zijn niet in te toetsen. De bytes 32 t/m 255 kunnen gePRINT worden, zonodig met expansie. Slechts indien er in een BASIC-programma werd gePOKEt of gePRINT (na OPENing) kunnen er uiteraard toch andere bytes in voorkomen.

E H F    Weijsers      -      Wilhelminalaan 42      -      2625 KH    Delft



In de vorige IMPULS-jaargang (72-09 & 73-27) schreef Wim Beekman twee artikelen over enkele hard- en softwarezaken betreffende de VTX5000. De softwarematige kant betrof met name het programmeren van de USART 8251A, het feitelijke hart van het VTX-interface.

Helaas heeft Wim niet meer een artikel geschreven waarin hij de theorie met voorbeelden heeft toegelicht. In dit artikel wil ik daarom een aantal zaken van een praktische kant benaderen. Ik ga ervan uit dat de lezer de genoemde IMPULS-artikelen gelezen heeft en het VTX-RS232-interface volgens Wims aanwijzingen verbouwd, dus met een schakelaar om verschillende baudrates te selecteren.

Met die baudrates wil ik beginnen. De instelling van de baudrate moet zowel hard- als softwarematig gebeuren. (Dat kon u al lezen in IMPULS 74-07.) In het 'mode register' bepalen de bits 1 en 0 de baudratefactor. Er zijn aldus vier mogelijkheden:

- 00 interface werkt in 'synchrone mode' (niet van toepassing)
- 01 baudrate volgens schakelaar x 64
- 10 baudrate volgens schakelaar x 4
- 11 baudrate volgens schakelaar

- Baudratefactor "01" werkt op het VTX-interface niet goed, want de ondersteunende hardware is niet aanwezig.
- Op de interfaceprint is 2400 de hoogst aangegeven baudrate, maar met baudratefactor "10" wordt dat ( $2400 \times 4 =$ ) 9600 baud.
- Met factor "11" werkt het interface op de ingestelde baudrate.

Bij bestaande modemsoftware wordt factor "11" gebruikt, want het oorspronkelijke VTX-modem vereist dat. Als met de schakelaar een andere baudrate gekozen wordt zal diezelfde software vrijwel altijd werken op de tegenwoordig populaire 2400 baud Hayes-modems. Factor "10" komt van pas bij printers en nulmodem-communicatie.

In het artikel van Wim Beekman wordt duidelijk dat de USART ook in BASIC te programmeren is. Dat klopt inderdaad, maar heb niet al te hoge verwachtingen van de snelheid, ook niet op 9600 baud. Ook voor het besturen en testen van de handshake-lijnen is BASIC te traag. Zenden lukt meestal wel, maar ontvangen niet altijd.

In de volgende BASIC-regels heb ik voor de bit-operaties gebruik gemaakt van de BIN-functie. Met de eerder genoemde IMPULS-artikelen ernaast is de werking dan waarschijnlijk beter te volgen.

- 2 <---- 3 De BASIC-regels zijn getest met twee Spectrums,
- 3 ----> 2 verbonden door een "nulmodem" (zie hiernaast).
- 4 <---- 5 Op de tweede Spectrum werkte FIDOTERM, op 1200/
- 5 ----> 4 1200. Ik gebruik baudratefactor "10", daarom
- 7 ----- 7 moet mijn baudrateschakelaar op 300/300 staan.

De door Wim beschreven VTX-ROM-schakelaar staat altijd op "uit".



Eerst moeten we het interface initieren:

```

10 FOR f=1 TO 3:           ; stuur 3 * 32 om zeker in het
    OUT 255,BIN 00100000:   ; 'command register' te werken.
    NEXT f:
11 OUT 255,BIN 01100000:   ; naar 'mode register'.
12 OUT 255,BIN 11001110:   ; 2 stopbits, geen parity,
                           ; 8 bits, baudratefaktor "10".
13 OUT 255,BIN 00110111   ; reset foutbits, RX enable,
14 STOP                    ; maak DTR hoog, TX enable.

```

Regels 10, 11 en 13 beschrijven het 'command register'. Bit 5 is daarbij steeds "1". Dat is hier zo niet belangrijk, maar wel als we straks dezelfde instructies in MC gaan gebruiken. We kunnen ermee voorkomen dat de VTX-ROM ongewild ingepaged wordt. Regel 12 beschrijft het 'mode register'. Regel 13 maakt DTR constant hoog. Voor FIDOTERM hoeft dit niet, maar andere terminalprogramma's verwachten dat misschien wel.

Het versturen van bytes kan zo:

```

20 LET b=IN 255:           ; in het 'status register' wordt
    IF b<>INT (b/2)*2 THEN ; bit 0 getest ("1" = TX ready).
    PAUSE 0:               ; elke toetsaanslag
    OUT 127,CHR$ INKEY$:   ; wordt verzonden.
    IF INKEY$=CHR$ 13 THEN ; na CHR$ 13 sturen we tevens
    OUT 127,10             ; een CHR$ 10.
21 GO TO 20

```

Zoals gezegd zijn deze regels getest met FIDOTERM op een tweede computer. Aangezien modemsoftware niet volgens het 'handshake'-protocol werkt hoeven we niet te testen of DSR 'hoog' is. Wel is een test opgenomen of onze USART 8251 klaar is om te OUTputten. Alle verzonden tekens worden door FIDOTERM op het scherm gezet.

Het ontvangen van bytes kan zo:

```

30 LET b=IN 255:           ; in het 'status register' wordt
    IF b>INT (b/4)*4+1 THEN ; bit 1 getest ("1" = RX ready).
    PRINT CHR$ IN 127;      ; het gelezen byte wordt gePRINT.
31 GO TO 30

```

Wederom geen handshake-opdrachten, maar wel een test of de USART een byte heeft ontvangen. Als dit zo is, dan wordt het byte met een INput-instructie gelezen en op het scherm gezet. Dat byte is dus een door FIDOTERM verstuurd toetsaanslag. Simpel eigenlijk.

Het principe van ontvangen en versturen is hiermee voldoende uit de doeken gedaan. Met IN 255 kunnen we het 'status register' lezen en via poort 127 kunnen we een byte INputten of OUTputten.



Het is natuurlijk handiger om het VTX-RS232-interface geheel in BASIC te besturen. De onderstaande HEXDATA-regels bevatten de MC voor een VTX-'device driver'. Over het gebruik leest u verderop.

DE MC "C5000">>"CODE 65300,202 IN HEXDATAREGELS:

1	"2A 4F 5C 11.0F 00 19 11.3D FF 73 23.72 23 11 7C"	1043
2	"FF 73 23 72.06 03 3E 20.D3 FF 10 FA.3E 60 D3 FF"	3021
3	"3E CE D3 FF.3E 35 D3 FF.C9 CD D3 FF.20 27 FE A5"	5698
4	"38 05 D6 A5.C3 10 0C FD.CB 01 86 FE.7F 38 02 3E"	7453
5	"3F FE 0D 20.07 CD 69 FF.3E 0A 18 09.FE 20 D8 20"	9026
6	"04 FD CB 01.C6 47 CD 54.1F D2 00 0D.DB FF E6 85"	11136
7	"FE 85 20 F2.78 D3 7F C9.FD CB 02 9E.FD CB 37 6E"	13693
8	"28 17 ED 7B.3D 5C D1 D1.ED 53 3D 5C.CD 9D FF 30"	15825
9	"FB FE 0D C8.CD 85 0F 18.F3 DB FF CB.4F 20 24 CD"	18192
10	"54 1F D2 00.0D F3 21 80.3E 3E 37 D3.FF DB FF CB"	20256
11	"4F 20 0C 2B.7C B5 20 F5.3E 35 D3 FF.AF FB C9 3E"	22274
12	"35 D3 FF DB.7F CD D3 FF.20 02 CB BF.37 FB C9 47"	24816
13	"3A DD FF F6.20 FE 74 78.C9 62"	26417

De initiatie zal u wellicht bekend voorkomen:

```
CLEAR 65299; LOAD "Cvtxrs232"CODE 65300; RANDOMIZE USR 65300
```

Daarna verwijst het altijd aanwezige "P"-channel niet meer naar routines voor de ZX Printer, maar naar bovenstaande routine (die nu dus gekoppeld is aan stream 3). Met deze routine kunt u zowel zenden als ontvangen. Verder kunt u instellen of de routine een "b"- of "t"-channel moet emuleren (nabootsen). Dat doet u met:

```
POKE 65501,CODE "t"           of           POKE 65501,CODE "b"
```

De BASIC-opdrachten met LPRINT, LLIST, INPUT #3 en INKEY\$#3 zijn te gebruiken. Deze ENE routine heeft nu alles wat u nodig heeft: zowel IN- als OUTPUT-routines, werkend als "b"- of "t"-channel.

Verder werken ook specifieke Opus-opdrachten. Zo kan ik geven: MOVE 1;"filenaam" TO "p". Daarmee wordt een file van disk over de VTX-RS232-poort verstuurd, bv naar een andere computer. Met OPEN #4;"b": MOVE #3 TO #4 maak ik mijn eigen serieel-parallel-converter. Andersom kan natuurlijk ook. Ik weet niet of dit ook bij andere disksystemen werkt. Het lijkt me wel waarschijnlijk.

De routine werkt met het 'handshake'-protocol. Dwz dat VOOR het verzenden gecontroleerd wordt of de andere partij kan ontvangen (of DSR 'hoog' is). En VOOR het ontvangen wordt kenbaar gemaakt dat de andere partij kan gaan zenden (DTR wordt 'hoog' gemaakt).

Op de volgende bladzijde staat de complete assembly-listing. Let vooral op de routines INIT2, TXBYTE en RXBYTE (incl handshake).

Kees Versluis - Copernicuslaan 25 - 2561 VA Den Haag



```

ERR_SP EQU 23613
CHANS EQU 23631
BREAK EQU $1F54

INIT LD HL,(CHANS) ; "P"-channel verwijst
LD DE,15 ; naar nieuwe routines
ADD HL,DE
LD DE,OUTBYT ; OUTPUT-routine
LD (HL),E
INC HL
LD (HL),D
INC HL
LD DE,IN_BYT ; INPUT-routine
LD (HL),E
INC HL
LD (HL),D

INIT2 LD B,3 ; stuur 3 x CHR$ 32
INIT3 LD A,$00100000
OUT ($FF),A
DJNZ INIT3
LD A,$01100000 ; naar mode register
OUT ($FF),A
LD A,$11001110 ; 8 bits, geen parity,
OUT ($FF),A ; 2 stopbits, factor 10
LD A,$00110101 ; reset foutbits, RX- en
OUT ($FF),A ; TX-enable, DTR laag
RET

OUTBYT CALL T_CHAN ; "b"- of "t"-channel?
JR NZ,TXBYTE

FILTER CP $A5 ; token-expansie, etc.
JR C,NO_TOK
SUB $A5
JP $0C10

NO_TOK RES 0,(1Y+1)
CP $7F
JR C,NOGRAF
LD A,"?"
NOGRAF CP $0D
JR NZ,NOT_CR
CALL TXBYTE
LD A,$0A
JR TXBYTE
NOT_CR CP $20
RET C
JR NZ,TXBYTE
SET 0,(1Y+1)

TXBYTE LD B,A ; bewaar A
TEST CALL BREAK ; BREAK gedrukt?
JP NC,$0D00 ; dan foutmelding
IN A,$FF ; lees statusregister
AND $10000101 ; bekijk DSR, TX-EMPTY en
CP $10000101 ; TX-READY
JR NZ,TEST
LD A,B ; herstel A
OUT ($7F),A ; verstuur A
RET

IN_BYT RES 3,(1Y+2) ; Mode niet gewijzigd
BIT 5,(1Y+55) ; INPUT # of INKEY#?
JR Z,RXBYTE

INPUT LD SP,(ERR_SP) ; haal een erroradres
POP DE ; van de stack
POP DE
LD (ERR_SP),DE
INPUT2 CALL RXBYTE ; lees byte
JR NC,INPUT2 ; succes?
INPACC CP $0D ; is het een CR?
RET Z ; dan klaar
CALL $0FB5 ; voeg toe aan string
JR INPUT2

RXBYTE IN A,$FF ; lees statusregister
BIT 1,A ; RX-READY?
JR NZ,READ2 ; dan lezen
CALL BREAK ; BREAK gedrukt?
JP NC,$0D00 ; dan foutmelding
DI
LD HL,16000 ; test 16000 maal
LD A,$00110111 ; maak DTR hoog
OUT ($FF),A
TEST2 IN A,$FF
BIT 1,A ; RX-READY?
JR NZ,READ1 ; verlaat de loop
DEC HL
LD A,H
OR L
JR NZ,TEST2 ; test opnieuw
LD A,$00110101 ; maak DTR laag
OUT ($FF),A
XOR A ; reset Carry (geen succes)
EI
RET

READ1 LD A,$00110101 ; maak DTR laag
OUT ($FF),A
READ2 IN A,$7F ; lees byte
CALL T_CHAN ; "b"- of "t"-channel?
JR NZ,RX_END
RES 7,A
RX_END SCF ; set Carry (succes)
EI
RET

T_CHAN LD B,A ; bewaar A
LD A,(STATUS) ; maak kleine letter
OR $00100000 ; "t"-channel? set Zero-flag
CP "t" ; herstel A
LD A,B
RET

STATUS DEFB "b" ; emuleer "b"- of "t"-channel

Deze routine is uitgebreid getest in nulmodemsituaties
t/m 38400 baud. Ja, u leest het goed: 38400 baud!! Hoe
dan wel? Het antwoord krijgt u in de volgende IMPULS.

```



Een diskdrive bezit een jumper (verplaatsbaar stekkertje), die de drive aan een der driveselectlijnen A, B, C en D koppelt. Bij een standaard-OD zit drive A links en, indien aanwezig, drive B rechts. De OD kan geen drives via de lijnen C en D selecteren, tenzij dit toch mogelijk werd gemaakt door Victors hardware-aanpassing.

	A	B	C	D
1	P	-	-	-
2	-	P	-	-
3	S	-	-	-
4	-	S	-	-

Drivenummers worden gebruikt in BASIC-opdrachten zoals bv CAT 2, SAVE \*1;"run", OPEN #3;NS enz, maar eveneens in MC-instructies.

Welke van de drives gaat draaien bij gebruik van een drivenummer wordt bepaald in de bijbehorende disksubtabel in het IC6116 (zie IMPULS 72-16 en de informatie verderop over negen bytes daarin). De initiele instelling (die na inschakeling of USR 14070 vanuit de OD-ROM in die tabellen is gekopieerd) is in het bovenstaande schema aangegeven. De betekenis van P(rimair) en S(ecundair, met "Insert disk -, then press a key") is elke OD-gebruiker bekend. Hierdoor kan in iedere drive worden gewerkt met meerdere disks, die zelfs verschillend geFORMAT mogen zijn (40/80 tr, SS/DS ed), aangezien de gegevens daarover in aparte disktabellen staan. Bij de OD kunnen de drivenummers dus opgevat worden als disknummers.

Dmv het volgende programma kan de instelling in de disktabellen gewijzigd worden. Dit is vooral nuttig als u drie of vier drives wilt gebruiken, maar het werkt zelfs bij een enkele echte drive, echter alleen vanaf OD-ROM-versie 2.2 en als IC6116 aanwezig is. SAVE uw drive-instelling voor later gebruik met CODE 8192,2E3. De laatste 128 bytes van IC0 zijn overbodig (de ankerblokkopie). 2E3 ipv 2048 scheelt een blok en dus telkens tijd bij herLOADen.

#### HANDLEIDING BIJ HET INSTELPROGRAMMA "odrinst"

Eerst wordt u gevraagd aan welke lijnen er drives moeten kunnen zitten en aan welke nummers die primair toegekend moeten worden. Antwoord hier met nummer 0 voor een lijn waaraan geen drive zit. Daarna wordt bij ieder nog vrij nummer gevraagd welke dan reeds primair gekoppelde lijn er secundair aan toegekend moet worden.

Een primair nummer kan nul tot drie secundaire nummers bezitten. Gebruik de lijnen C en D niet voor primaire nummers indien uw OD daarvoor niet geschikt gemaakt is (zie het artikel van Victor). Sluit ook geen drives aan op de lijnen waaraan volgens uw opgave geen drives zitten, aangezien er anders meerdere drives tegelijk kunnen gaan draaien, wat tot de melding "Disk I/O error" leidt.

Wanneer u zich hebt vergist, BREAK het programma dan niet, maar kies 1 (REPEAT) bij de slotvraag, waarna u alles over kunt doen. (PRINT USR 14070 geeft u desnoods de initiele instelling terug).



HET BASICPROGRAMMA "odrinst" LINE 1 - 301090 - EdW -

```

1>DEF FN I()=CODE INKEY$#3+CODE INKEY$#3*256:
  DEF FN M$(A$)=CHR$ ((A$(1)>O$)+(2 AND A$(2)>O$)+
    (4 AND A$(3)>O$)+(8 AND A$(4)>O$))

5 CLEAR #: POKE23658,8: CLEAR : DIM A(6): DIM N$(4,4):
  LET D$="ABCD": LET O$="": LET P$=O$: LET E$="":
  LET A=1: LET B=2: LET C=3: LET D=4: GO SUB 55

10 OPEN #3;"CODE ":POINT #3;FN I()+10:POINT #3;FN I()
11 LET N=CODE INKEY$#3:
  IF N THEN
    LET A(N)=FN I()+2: GO TO 11
12 LET A(5)=FN I()+4

20 FOR L=A TO D
21 INPUT "PRIMARY ( INPUT 0 TO CONTINUE )",,,
  " LINE "+D$(L);" TO NUMBER ";N
22 IF N<1 OR N>4 THEN
  LET E$=E$+CHR$ L: GO TO 25
23 IF N$(N)<>O$ THEN GO TO 21
24 LET N$(N,L)="P": LET P$(L)=STR$ N: GO SUB 50
25 NEXT L
26 POINT #3;A(5): LPRINT FN M$(P$);:
  IF P$=O$ THEN RUN

30 FOR N=1 TO 4:
  IF N$(N)<>O$ THEN GO TO 34
31 INPUT "SECONDARY '"NUMBER "+N;" TO LINE ";L:
  IF L<A OR L>D THEN GO TO 31
32 IF P$(L)=" " THEN GO TO 31
33 LET N$(N,CODE E$)=CHR$ 128:
  LET N$(VAL P$(L),CODE E$)=CHR$ 128:
  LET N$(N,L)="S": LET E$=E$(2 TO): GO SUB 50
34 NEXT N

40 LET O$="S"
41 FOR N=1 TO 4:
  POINT #3;A(N)+6: LPRINT FN M$(N$(N));: NEXT N
42 INPUT "0 OK / 1 REPEAT'"N: CLEAR #: GO TO 60 OR N

50 POINT #3;A(N): LPRINT FN M$(N$(N));
55 PRINT ,AT 6,14; " LINE ";
  TAB 12;"N ";D$ ;" N";
  TAB 12;"U1";N$(1);"1U";
  TAB 12;"M2";N$(2);"2M";
  TAB 12;"B3";N$(3);"3B";
  TAB 12;"E4";N$(4);"4E";
  TAB 12;"R ";D$ ;" R";
  TAB 14; " LINE ": RETURN

```

een inverse rand  
om het vierkant  
van 4x4 maken mbv  
INV en TRUE VIDEO



# NEGEN BYTES MBT DE KOPPELING VAN NUMMER EN DRIVE IN HET IC6116

## ALGEMEEN:

- MASKER-0 (de High-nibble is 0) op adres:  
4 + pointer achter de disk-info-tabel
- DCBA wijst aan welke driveselectlijnen er  
00000011=3 gebruikt kunnen worden (initieel A en B)

## VOOR ELK VAN DE DRIVENUMMERS 1 T/M 4:

- MASKER-1 Low-nibble (High-nibble soms 0) op adres:  
2 + pointer achter het drivenummer  
in de disk-info-tabel (alle bits 1)
- Primair drnr 1 bit 1 om de driveselectlijn aan te wijzen
- Secundair drnr nog 1 extra bit 1 (geen driveselectlijn);  
bij meerdere secundaire nummers moeten deze  
extra bits op verschillende plaatsen staan
- AND MASKER-0 wijst de driveselectlijn bij een nummer aan
- MASKER-2 Low-nibble (High-nibble is 0) op adres:  
6 + adres MASKER-1
- Primair drnr alle extra bits der bijbehorende secundaire  
nummers 1 (dus 0 zonder secundaire nummers)
- Secundair drnr 1 bit 1 waar MASKER-1 het extra bit 1 heeft

## OPMERKINGEN

De High-nibble van MASKER-1 mag gerust 0 gemaakt worden, aangezien deze diskgegevens bevat die toch telkens uit het ankerblok worden overgenomen. Bit 6 en 7 bevatten de blok grootte en bit 4 geeft SS/DS aan. Bit 5 voor DD/SD wordt niet telkens bijgewerkt, maar waarschijnlijk gebruikt toch niemand SD (bit 5 is dan 1).

De PIA onthoudt het laatste MASKER-1 op adres 12288 (3000 hex), om te kunnen vaststellen of er een 'prompt' gegeven moet worden.

Opdat een geSAVEde IC6116-inhoud te herLOADen is mag er tijdens dat herLOADen geen wisseling van driveselectlijn optreden. Wisseling van drivenummer veroorzaakt slechts een Insert-boodschap.

E H F Weijgers - H Marsmanlaan 29 - 2624 TJ Delft



Deze pagina ziet er niet hetzelfde uit als de rest van deze Impuls, dat moet u wel opgefallen zijn. Nou zie ik u raden: "Is het met een laser- of 24naalds printer gemaakt?". Nee! "Met een inktjetprinter dan?". Nee, deze pagina is gemaakt op een 'good old' Star SG-10!

"En de paginaopmaak? Zeker met een PC gedaan?". Nee! "Of met een Mac of ST?". Nee! "Met een QL dan?". Ook niet! Deze pagina is gemaakt met het Cardex Desk Top Publishing pakket op een ZX Spectrum 48K.

*Graphics*

POG adverteert al geruime tijd in o.a. Your Sinclair met dit DTP-pakket, dat uit de volgende programma's bestaat:

- WordMaster, een tekstverwerker,
- Headliner, een tekenprogramma voor koppen of tekeningen.
- Typeliner, de eigenlijke Desk Top Publisher.

Laten we eens kijken wat de programma's te bieden hebben voor de 37,80 pond dat ze kosten.

### WordMaster

Dit is de tekstverwerker, maar ook de basis voor Headliner en Typeliner. Het programma is ongeveer 11K groot, er resteert dus een vrije ruimte van 29K.

Nadat WM geladen is, wordt de File Handling (FH) actief. Deze kan werken met cassette en elk diskinterface met Microdrivesyntaxis. Wel bestaan er kleine verschillen tussen deze diskversies.

Onderin het scherm staan (in 64 tekens per regel) de commandotoetsen vermeld. Deze bieden o.a. mogelijkheid tot het creëren, activeren, save, loaden, deleten en koppelen van files.

Door een file te creëren wordt WordMaster gestart. WM kent een krachtige 'search & replace' (beter dan TW3), een fast scroll, insert mode, blockdelete, -move, -copy, -save (!) en een onscreen underline. Een nieuwe ontwikkeling voor mij was dat er slechts bij het einde van een alinea een carriage return mag staan. Er moet dus meer met de pijltoetsen gewerkt worden dan met <enter>.

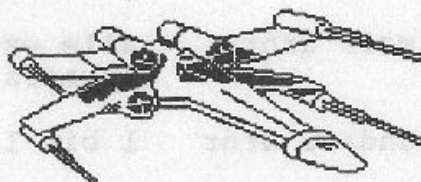
De 'word wrap' is automatisch en met 'insert mode on' wordt de paragraaf bij het schrijven steeds razendsnel geherformeerd.

Ook nieuw waren de 'commandlines'. Op deze regels, die anders gekleurd zijn dan de tekstregels,

staan tekst herformeringscommando's, waarvan het effect slechts bij het uitprinten of in Typeliner te zien is. Zo is er bijvoorbeeld 'fill', die de hele regel uitlijnt, of 'form', die een 'pagebreak' forceert. Enkelen zijn specifiek om te printen met WM, anderen werken ook in Typeliner.

Eigenlijk is WM alleen niet zo interessant. Alhoewel deze tekstverwerker ook graphics heel gemakkelijk tussen de tekst kan printen, kan ik hem eigenlijk niet aanraden voor slechts tekstverwerking.

Interessant wordt het pas als u ook Headliner en Typeliner heeft. Het unieke van WM (of eigenlijk FH) is dat er meerdere files tegelijk in het



geheugen aanwezig kunnen zijn, of dit nu teksten, fonts, graphics of pagedocumenten zijn. 'List' geeft hier een lijst van.

'Get' activeert een file. Is dit van het type tekst, dan wordt WM gestart en kun je de tekst veranderen. Is dit een extentieprogramma, dan wordt dit extentieprogramma gestart. Voorbeelden van dit soort programma's zijn Headliner en Typeliner. Zij kunnen niet zelfstandig in het geheugen werken, maar alleen in combinatie met WM.

Verder worden een fonteditor-, conversie-, backup- en meer extentieprogramma's meegeleverd.



## Headliner

Headliner (HL) is het tekenprogramma. Het wordt door FH geload. Met <G> en hierna 'headliner' wordt het geactiveerd.

HL biedt plot, draw, circle, arc, fill, enlarge (gedetailleerd tekenen), een simpele undo en tekst tekenen met een van de zes meegeleverde gigantische fonts (zie plaatje hieronder) of met eigen (standaard) fonts (die overigens eerst omgezet moeten worden). Cardex heeft ook nog meer fonts en clip art in de aanbieding.

De 4092 bytes grote HL is natuurlijk geen Art Studio, maar men kan er voldoende mee doen: 'It gets the job done.' Natuurlijk kunnen SCREEN\$s, gemaakt met Art Studio of Artist, geïmporteerd worden.

Een graphic, een (deel van een) screen\$, wordt opgeslagen door een file te creëren. U moet dan aangeven op het scherm welk deel u wilt opslaan door een vierhoekige omlijning te positioneren. Deze graphics zijn dan in Typeliner te gebruiken.

# Outline! RO MAN!

## Typeliner

Dit programma wordt, net als HL, door FH geload en gestart met 'Get'. TL is het pagina layout programma van het DTP pakket. Meerdere pagina's kunnen in de FH aanwezig zijn, maar TL kan slechts EEN pagina tegelijk aan.

Dit programma was de grootste verrassing voor mij. TL start op door naar een (bestaande) tekstfile te vragen. Hierna krijg je de pagina (A4 formaat) te zien waarop het allemaal gaat gebeuren.

De pagina kan maximaal bevatten: 12 graphics (van willekeurige grootte), 7 fonts, 24 lijnen/omlijningen in 6 stijlen en 24 blokken om tekst te positioneren. Geloof me, dit is meer dan voldoende om een professioneel uitziende pagina te produceren.

De volgende fonts worden meegeleverd: R-light,

R-bold, R-heavy, C-light, C-bold, Uniclight, Scrawl, City, Title en Mini. Deze zijn slechts in TL te gebruiken maar produceren op papier Near Letter Quality!

De gemaakte pagina is te bekijken door de 'view'-opdracht. De hele pagina wordt dan op scherm vertoond, alles natuurlijk verkleind (zelfs graphics!). De tekst wordt met enkele pixels per letter getoond. Onleesbaar uiteraard, maar de tekstopzet (spaties ect.) is duidelijk te zien. Gelukkig is op de linkerheft van het scherm de tekst wel leesbaar geschreven, zodat je wel kan zien welke tekst in welk tekstblok begint.

Is de tekst niet op de juiste plaats? Even het tekstblok verplaatsen, weer 'view' en voila. Voor mij is dit gemak zo groot, dat ik voortaan alleen nog maar met dit DTP-pakket mijn handleidingen, verslagen, ect. zal gaan maken. Gelukkig zijn teksten van elke (?) andere tekstprocessor te loaden. U heeft dan wel werk aan het herformuleren van de alinea's.

TL biedt WYSIWYG (spreek uit: 'wizziwig'): What You See Is What You Get!! Zoals de pagina op het scherm staat, zo komt-ie ook op papier! Zelfs gehele SCREEN\$s komen prachtig op papier terecht (eventueel een shade copy!). Magnifiek toch?

De printkwaliteit is zeer hoog, doordat de printerkop twee keer over de graphics en drie keer over de tekst gaat (dus zeer uitputtend voor het lint!)

Typeliner werkt met matrixprinters die 1/216 inch linefeed kennen (ofwel fully EPSON compatible printers). Informeer voor alle zekerheid bij Cardex of uw printer geschikt is voor het DTP pakket. Meegeleverd worden wel speciale Typeliners voor oa de Brother HR-5 en enkele IBM-compatibele printers en natuurlijk een 'printercode customize' programma.

Ook is nog vermeldenswaard, dat eigen TL-fonts ontworpen kunnen worden met het extensieprogramma 'editor'. Ook kunnen fonts uit Qualitas+ omgezet worden naar Typeliner-fonts.



### Conclusie

#### Plus:

- een pakket dat professioneel uitziende pagina's produceert.
- geschikt voor vele printers.
- loopt al op een spectrum 48K.
- Versies voor vele opslaginterfaces.
- Het flexibele FileHandling systeem: geen programma's in het geheugen die zelden gebruikt worden.

#### Min:

- alle programma's steunen zwaar op het 'plus'-toetsenbord. Met name Inv./True video, delete, edit moeten veel worden ingedrukt.
- WM werkt wel handig met multiple files, maar de tekstverwerker heeft z'n onhebbelijkheden (zoals iedere andere overigens).
- een disk error doet je in BASIC belanden.

Voor iemand die met de Spectrum wilt gaan typesetten, is dit HET pakket. Professioneel, vol gemak en vooral MOOL.

Het gemak waarmee tussen programma's gewisseld kan worden, is uniek voor de Spectrum. Op deze manier biedt dit systeem vele mogelijkheden voor uitbreidingen.

Het DTP-pakket is voor 37,80 pond (voor enkele diskversies geldt een toeslag) verkrijgbaar bij

**PCG,  
Barrow-in-Furness,  
Cumbria,  
LA14 1EW, Great Britain.**

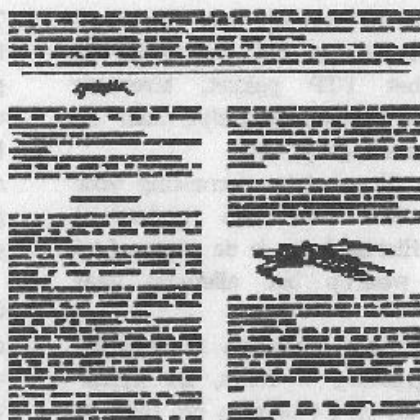
Vermeld wel welk type spectrum, diskinterface en printer u heeft.

(Victor Vogelpoel)



### Text .DTPJordan.1

```
Block:01
Style:02
>fill!
Deze pagina ziet er
niet hetzelfde uit
als de rest van deze
Impuls, dat moet u
wel opgevalen zijn.
Nou zie ik u raden:
"Is het net een
laser- of 24naalids
printer genaakt?".
Nee! "Niet een
inktjetprinter dan?".
Nee, deze pagina is
genaakt op een "good
old" Star 56-10!
"En de paginaopmaak?
Zeker niet een PC
gedaan?". Nee! "Of
```



Plaatje: Schermweergave door Typeliner van pagina 1 van deze review.



In IMPULS 82-21 stond het MERGE-programma "Xpar>ser", waarmee u in het IC 6116 een routine kon installeren voor het Opus-RS232-interface. Die routine bevat helaas een foutje in het ontvangst-gedeelte, dat zich alleen openbaarde bij lagere baudrates. Door-dat ik de afgelopen tijd ook andere RS232-interfaces (oa VTX5000 en VTX711) getest heb kwam de fout aan het licht. Op de volgende pagina staat de verbeterde versie, die ik voor de duidelijkheid "Xpar>ser+" genoemd heb. Met gemeente excuses voor het ongemak.

Als u al beschikt over de vorige versie van dit MERGE-programma, dan behoeft in het eerste deel alleen regel 63 gewijzigd te worden (m+104 wordt m+106). De DATA-regels daarna moeten bijna alle worden overgenomen. Downloaden uit "Sinclair Box" kan ook. Samen met Victors baudrate-opdracht in deze IMPULS beschikt u nu over prima gereedschap voor RS232-communicatie tot en met 19200 baud.

Ik maak meteen van de gelegenheid gebruik om nog wat opmerkingen te maken over deze seriele routine. Ten eerste de initiatie. Het interface wordt telkens opnieuw geïnitieerd, dwz telkens als er een byte verstuurd of ontvangen wordt. Het betekent echter ook, dat de 5 volt (pull up) op pen 9 er pas is nadat de routine eenmaal is aangesproken voor zenden of ontvangen.

In mijn vorige stukje (IMPULS 82-21) schreef ik dat elke normale OD-versie van een programma zou werken met deze seriele routine. De genoemde voorbeelden (BETA BASIC, TW2, TW3, SW, Devpac) kloppen. Daarin moet eerst stream 3 worden geopend naar het "t"- of "b"-channel, waarna steeds de 'output' naar stream 3 gaat. Ook de meeste screendumps zullen het wel doen, omdat ze eveneens stream 3 gebruiken of omdat ze (keurig netjes) CALPHY aanroepen.

Uitzonderingen zijn de twee ART-STUDIO-conversies van Marcel van Dongen. In zijn 'printer driver' wordt de 6821 PIA direct geprogrammeerd voor parallel gebruik. De oplossing ligt voor de hand: een 'printer driver' die CALPHY aanroept en dus altijd werkt. Op een volgende DUCDISK komen de AS-conversies, incl deze 'driver'. Tot die tijd mag u me er altijd om vragen (in Houten) of bellen.

De manier waarop deze seriele routine aan het 'operating system' van de Opus is vastgeknoopt kan in principe ook worden toegepast voor elk ander parallel of serieel interface. Voor eigen gebruik heb ik een 6116-CODEblok met een routine voor de VTX5000-RS232, dat ik (zonodig) kan LOADen. Mijn VTX-interface is altijd aangesloten, dus het LOADen van een 6116-CODEblok is minder werk dan het aansluiten van een ander interface. Als daar belangstelling voor is wil ik die VTX-routine wel eens in dit blad beschrijven.



HET MERGE-PROGRAMMA "Xpar>ser+"

```

60 POINT #3;8192: LET p=FN g()+8:
   POINT #3;p: LET t=FN g():
   POINT #3;t: LET t$=""
61 LET t$=t$+INKEY$#3+INKEY$#3+INKEY$#3:
   IF CODE t$(LEN t$-2) THEN GO TO 61
62 FOR f=1 TO LEN t$ STEP 3:
   IF t$(f)<>CHR$ 129 THEN NEXT f
63 POINT #3;t+f: LET s=FN g():
   RANDOMIZE m: LET s$=FN g$():
   POINT #3;s+4: RANDOMIZE m+106:
   LET s$=s$+FN g$()+INKEY$#3+INKEY$#3+INKEY$#3+INKEY$#3
64 IF s<8192 THEN LET s=m+L: LET L=L+LEN s$
65 IF t<8192 THEN LET t=m+L: LET L=L+LEN t$
66 IF m+L>c AND m=b OR m+L>e THEN
   PRINT "MC te lang": CLEAR #: GO TO 3
67 POINT #3;s: LPRINT s$;:
   RANDOMIZE s: LET t$(f+1 TO f+2)=FN g$()
68 POINT #3;t: LPRINT t$;:
   POINT #3;p: RANDOMIZE t: LPRINT FN g$();
69 POINT #3;10095:
   LPRINT CHR$ 12+CHR$ 0;CHR$ 12+CHR$ 0;CHR$ 2;

100 327 : REM DATABLOK tbv MC - DATA achter elk regelnummer
101 "7C 06 0B 2F.4F 3A 73 27.D3 FE DD 21.02 30 DD 36",1523
102 "01 00 DD 36.00 0F DD 36.01 04 DD 36.00 08 2A 6F",2530
103 "27 2B 54 5D.1B 1B 1B 7A.B3 20 FB CD.5B 14 DD CB",4194
104 "00 76 28 F7.37 F3 DA A0.EA DD CB 00.96 C3 A7 EA",6679
105 "DD CB 00 D6.C3 A7 EA 54.5D 1B 7A B3.20 FB 3E 00",8763
106 "AF CB 39 10.E1 FB DD 36.00 08 2B 7C.B5 20 FB 3A",10662
107 "48 5C E6 38.0F 0F 0F D3.FE C9 21 A5.EB 7E A7 28",12589
108 "06 36 00 23.7E 37 C9 F3.ED 5B 71 27.21 20 03 42",13923
109 "4B CB 38 CB.19 CD 5B 14.DD 21 02 30.DD 36 01 00",15381
110 "DD 36 00 0F.DD 36 01 04.DD 36 00 0A.DD CB 00 6E",16770
111 "28 12 DD CB.00 6E 28 0C.DD CB 00 6E.28 06 DD CB",18418
112 "00 6E 20 0C.2B 7C B5 20.E3 DD 36 00.08 F5 18 2E",19777
113 "60 69 3A 73.27 D3 FE 06.80 2B 2B 2B.2B 2B 19 2B",21072
114 "2B 7C B5 20.FB DD CB 00.6E CA 3F EB.37 18 03 B7",23002
115 "18 00 CB 18.30 E8 DD 36.00 08 78 2F.37 F5 19 2B",24351
116 "7C B5 20 FB.19 19 19 2B.7C B5 28 43.DD CB 00 6E",26003
117 "28 F5 DD CB.00 6E 28 EF.DD CB 00 6E.28 E9 DD CB",28332
118 "00 6E 28 E3.62 CB 3C 69.06 80 2B 2B.2B 2B 19 2B",29549
119 "2B 7C B5 20.FB DD CB 00.6E CA 8F EB.37 18 03 B7",31559
120 "18 00 CB 18.30 E8 21 A5.EB 36 01 23.78 2F 77 CD",33104
121 "BF EA F1 FB.C9 00 00" ,34222
122 "",0 : REM DATABLOK tbv opdracht
123 55,64,62,71,69,71,107,325,218,223,298,303
124 311,325,320,95,0 : REM DATABLOK tbv plaats
125 44,64,45,21,134,64,135,21,0 : REM DATABLOK tbv versie

```



02	COLOFON .....	--
03	VAN DE REDACTIE .....	--
04	THE EXTENDED ART STUDIO MET RAMDISK 6 .....	OD
05	HET VTX711-RS232-INTERFACE - DEEL 1 .....	SP
08	OPUS 2 - STREAMS EN CHANNELS, POINT EN FORMAT .....	OD
10	DE VTX5000-RS232 ALS MODEMINTERFACE .....	SP
11	VIER ECHTE DRIVES AAN DE OPUS-DISCOVERY .....	OD
14	EEN QUICKSORT VOOR STRINGARRAYS .....	SP
16	BLOKMOVE IN BASIC - FILL STRING .....	OD
18	DE INHOUD VAN EEN BETADISK - DEEL 2 .....	BD
20	DE EXTRA OPUSOPDRACHTEN LOAD @ & SAVE @ .....	OD
22	VIDEOTEX - EEN ANDERE VORM VAN DATACOMMUNICATIE? .....	--
26	DE EXTRA BASICOPDRACHT: FORMAT "B"/"T";BAUDRATE .....	OD
27	BD>OD - VAN BETADISK NAAR OPUSDISK .....	OD
31	REKENROUTINES - DEEL 2 .....	--
35	PROGRAMMA'S OM TEKST- EN BASICFILES TE LEZEN .....	OD
39	HET PROGRAMMEREN VAN DE VTX5000-RS232 .....	SP
43	DRIVENUMMERINSTELLING .....	OD
46	DTP OP EEN SPECTRUM .....	SP
49	VERBETERDE SOFTWARE VOOR HET OD-RS232-INTERFACE .....	OD
51	DE INHOUD VAN IMPULS 84 - DE SGG-DAGEN IN HOUTEN .....	--

-- ALGEMEEN	SP ZXSPPECTRUM	DD DISCIPLE
80 ZX80	MD MICRODRIVE	QL QUANTUM LEAP
81 ZX81	OD OPUS DISCOVERY	88 Z88
CR CASSETTERECORDER	BD BETADISK	PC SINCLAIR PC

# DE SGG-DAGEN IN HOUTEN

ZATERDAG	10-16 UUR	HCC-KANTOOR
19 JANUARI	16 MAART	18 MEI
22 JUNI	24 AUGUSTUS	26 OKTOBER

# DE HCC-MICROCOMPUTERDAGEN

VRIJDAG 29 NOVEMBER	ZATERDAG 30 NOVEMBER	1991
IN "DE JAARBEURS"		TE UTRECHT

ONDER VOORBEHOUD - BEKIJK STEEDS DE AGENDA IN DE HCC-NIEUWSBRIEF



SINCLAIR IMPULS

Postbus 76  
2260 AB Leidschendam

PORT BETAALD  
PORT PAYE  
DEN HAAG

SINCLAIR